UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICA CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

SISTEMA QUE SIMULE CONOCIMIENTO HUMANO CON VOZ “SEUZ”

DESARROLLO DE SOFTWARE

PROYECTO # 1

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: DARIO PATRICIO ALARCON ACOSTA

TUTOR: ING. XAVIER LOAIZA

GUAYAQUIL – ECUADOR

2011

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICA CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

SISTEMA QUE SIMULE CONOCIMIENTO HUMANO CON VOZ “SEUZ”

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

DARIO PATRICIO ALARCON ACOSTA

TUTOR: ING. XAVIER LOAIZA

GUAYAQUIL – ECUADOR

2011

MANUAL TECNICO

Tabla de contenido

MANUAL TECNICO 1
DICCIONARIO DE DATOS 3
Nombre de la tabla: Configuración 3
Nombre de la tabla: Categoria 3
Nombre de la tabla: Conocimiento 3
Nombre de la tabla: Palabras 4
Nombre de la tabla: Usuarios 4
Scripts de Tablas 5
Tabla Configuración 5
Tabla Categoria 5
Tabla Conocimiento 5
Tabla Palabras 6
Tabla Usuarios 6
DFD 8
Clase Voz 9
Clase usuarios 13
Pantalla temas 19
Clase reglas 32
Pantalla principal 35
Modulo 99
Ingreso 101
Diseño 103
Ingresar usuario 109
Clase conocimiento 121
Clase configuraciones 128
Pantalla bienvenida 134
Pantalla de Ayuda 139

Pantalla Aprender 143
MANUAL DE USUARIO 158
Pantalla de Bienvenida 159
Pantalla de Login 160
Pantalla principal 161
Modo1 161
Modo2 162
Pantalla de Guardar Como 162
Pantalla de Impresión 163
Pantalla de Tipo de letra 163
Modo 1 163
Modo 2 163
Pantalla de Color de Texto 164
Modo 1 164
Modo 2 164
Pantalla de Temas 165
Modo 1 165
Modo 2 165
Pantalla de Ingreso de Nuevo Usuario 166
Modo 1 166
Modo 2 166
Pantalla de Ingreso de Conocimiento 167
Modo 1 167
Modo 2 167

DICCIONARIO DE DATOS

Nombre de la tabla: Configuración

El objetivo de crear la tabla configuración es obtener y cambiar los estilos visuales del sistema como lo es el tema, tamaño de letra etc.

Campos	Tipo de Dato	Descripción
id	int(10)	pk.clave primaria de la tabla
tema	varchar(45)	nombre del tema
tama_Letra	int(10)	tamaño de la letra
tipo_letra	varchar(45)	nombre del tipo de letra
color	varchar(45)	color de la letra

Nombre de la tabla: Categoria

El objetivo de crear la tabla categoria es obtener el tipo de conocimiento que posee el sistema por ejemplo: batalla de pichincha esta en la categoria historia.

Campos	Tipo de Dato	Descripción
id	int(10)	pk.clave primaria de la tabla
descripcion	varchar(450)	nombre de la categoría
estado	varchar(1)	estado de la categoria

Nombre de la tabla: Conocimiento

El objetivo de crear la tabla conocimiento es obtener y guardar la informacación que posee el sistema ejemplo información de programación.

Campos	Tipo de Dato	Descripción
id	int(10)	pk.clave primaria de la tabla
tema	varchar(60)	nombre de la informacion
descripción	varchar(5000)	descripción del tema o contenido
id_usuario	int(10)	codigo del usuario que creo la informacion
fecha_cre	datetime	fecha de creacion la información
id_catagoria	int(10)	id de la categoria
fecha_mo	datetime	fecha de modificacion
fuente	varcha(1500)	fuente donde se obtuvo la información
observación	varchar(5000)	observacion de la información

estado	varchar(1)	estado de la información

Nombre de la tabla: Palabras

El objetivo de crear la tabla palabras es obtener los comandos que posee el sistema.

Campos	Tipo de Dato	Descripción
id	int(10)	pk.clave primaria de la tabla
descripcion	varchar(450)	nombre del comando
estado	varchar(1)	estado del comando

Nombre de la tabla: Usuarios

El objetivo de crear la tabla usuario es obtener y guardar la información del los usuarios que van a manipular el sistema.

Campos	Tipo de Dato	Descripción
id	int(10)	pk.clave primaria de la tabla
nombre	varchar(45)	nombres del usuario
apellido	varchar(45)	apellidos del usuario
clave	varchar(45)	clave de ingreso del usuario
user	varchar(45)	nick del usuario
estado	varchar(1)	estado del usuario

Scripts de Tablas

Tabla Configuración

```
DROP TABLE IF EXISTS `seuz`.`configuracion`;

CREATE TABLE  `seuz`.`configuracion` (

  `id` int(10) unsigned NOT NULL auto_increment,

  `tema` varchar(45) NOT NULL,

  `tama_Letra` int(10) unsigned NOT NULL,

  `tipo_letra` varchar(45) NOT NULL,

  `color` varchar(45) NOT NULL,

‘errores’ int(10)

  PRIMARY KEY  (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

Tabla Categoria

```
DROP TABLE IF EXISTS `seuz`.`categoria`;

CREATE TABLE  `seuz`.`categoria` (

  `id` int(10) unsigned NOT NULL auto_increment,

  `descripcion` varchar(450) NOT NULL,

  `estado` varchar(1) NOT NULL,

  PRIMARY KEY  (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Conocimiento

```
DROP TABLE IF EXISTS `seuz`.`conocimiento`;

CREATE TABLE  `seuz`.`conocimiento` (

  `id` int(10) unsigned NOT NULL,
```

```
  `tema` varchar(60) NOT NULL,
  `descripcion` varchar(5000) NOT NULL,
  `id_usuario` int(10) unsigned NOT NULL,
  `fecha_cre` datetime NOT NULL,
  `id_categoria` int(10) unsigned NOT NULL,
  `fecha_mo` datetime NOT NULL,
  `fuente` varchar(1500) NOT NULL,
  `observacion` varchar(5000) NOT NULL,
  `estado` varchar(1) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Palabras

```
DROP TABLE IF EXISTS `seuz`.`palabras`;

CREATE TABLE  `seuz`.`palabras` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `descripcion` varchar(45) NOT NULL,
  `estado` varchar(1) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

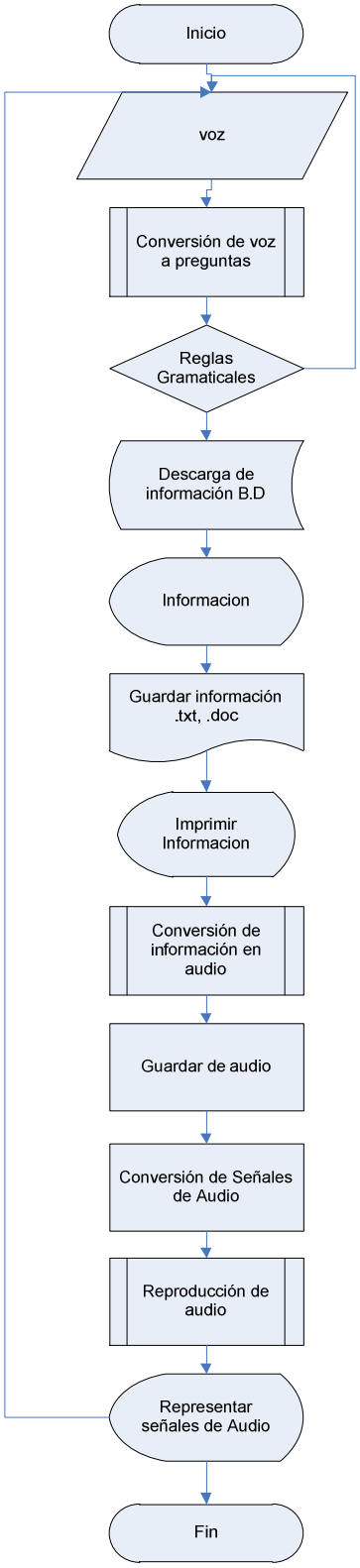
Tabla Usuarios

```
DROP TABLE IF EXISTS `seuz`.`usuarios`;

CREATE TABLE  `seuz`.`usuarios` (
```

```
  `id` int(10) unsigned NOT NULL auto_increment,
  `nombre` varchar(45) NOT NULL,
  `apellido` varchar(45) NOT NULL,
  `clave` varchar(45) NOT NULL,
  `user` varchar(45) NOT NULL,
  `estado` varchar(1) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

DFD


Inicio
voz
Conversión de voz a preguntas
Reglas Gramaticales
Descarga de información B.D
Informacion
Guardar información .txt, .doc
Imprimir Informacion
Conversión de información en audio
Guardar de audio
Conversión de Señales de Audio
Reproducción de audio
Representar señales de Audio
Fin

Clase Voz

Esta clase me permite cargar las voces que posee el sistema operativo

Librerías que me permiten sintetizar la voz y guardar archivos de audio

```
Imports SpeechLib //librería que me permite interactuar con las voces de loquendo
Imports System.Environment
Imports ACTIVEVOICEPROJECTLib
Imports System.Threading

Public Class voz
```

Atributo que me permite interactuar con una voz determinada

```
Private WithEvents pvoz As New SpVoice
```

Atributo que contiene la velocidad de la voz

```
Private pvelocidad As Integer = 2
```

Contiene todas las voces del sistema

```
Private pvoces As New ComboBox
```

Contiene la velocidad el volumen de la voz

```
Private pvolumen As Integer = 50
```

Atributo que me permite guardar la voz

```
Dim voz As New DirectSS
```

Método que me permite seleccionar la voz con la que va a trabajar el sistema

```
   Public Sub cargar ()
```

Cargo todas las voces del sistema

```
      Dim Token As ISpeechObjectToken
For Each Token In pvoz.GetVoices
pvoces.Items.Add (Token.GetDescription ())
      Next
```

Escojo la primera voz que tiene instalado el sistema operativo

```
pvoces.SelectedIndex = 0
```

Cargo los modos de la voz

```
Dim i As Integer
      Dim tipoVoz As String
      For i = 1 To voz.CountEngines
         tipoVoz = voz.ModeName(i)
      Next
   End Sub
```

Método que me permite transformar una cadena de caracteres en audio

```
Public Sub leer(ByVal cadena As String)
      Try
                Selecciono la primera voz del sistema
pvoz.Voice = pvoz.GetVoices().Item(0)
         Dim dt As DateTime
         dt = Now
            Establezco la velocidad de la voz
         pvoz.Rate = pvelocidad
                Establezo el volumen de la voz
         pvoz.Volume = pvolumen
                Preparo el dispositivo de audio
pvoz.Speak("".ToString, SpeechVoiceSpeakFlags.SVSFPurgeBeforeSpeak)
         Try
                Transformo la cadena de caracteres a audio
pvoz.Speak(cadena, SpeechVoiceSpeakFlags.SVSFPurgeBeforeSpeak)
         Catch ex As Exception
            MsgBox(ex.ToString, MsgBoxStyle.Exclamation, "I'm Speechless")
         End Try
      Catch ex As Exception
         Esta parte es la misma que la anterior pero función si la primer try dio error esto puede
         suceder si estamos trabajando en Windows server
```

```
Try
            voz.Speed = pvelocidad
            voz.VolumeLeft = pvolumen
            voz.Speak(cadena)
        Catch exx As Exception
            MsgBox(exx.ToString)
        End Try
    End Try
  End Sub

Método que me permite cambiar u obtener la propiedad del volumen
Public Property volumen() As Integer
' bloque Get para devolver
    ' el valor de la propiedad
Get
        Return pvolumen
    End Get
    ' bloque Set para asignar
' valor a la propiedad
    Set(ByVal Value As Integer)
pvolumen = Value
    End Set
  End Property
Método que me permite cambiar u obtener la propiedad de velocidad
  Public Property velocidad () As Integer
'bloque Get para devolver
    ' el valor de la propiedad
Get
        Return pvelocidad
    End Get
```

```
        ‘Bloque Set para asignar
‘valor a la propiedad
        Set(ByVal Value As Integer)
pvelocidad = Value
        End Set
    End Property

Método que me permite guardar una cadena de caracteres en un archivo de audio
Public Sub guardar(ByVal info As String, ByVal archivo As String)
        Try
                Preparo el dispositivo de audio
            Dim SpFlags As SpeechVoiceSpeakFlags = SpeechVoiceSpeakFlags.SVSFlagsAsync
            Dim SpFileMode As SpeechStreamFileMode =
SpeechStreamFileMode.SSFMCreateForWrite
            Dim SpFileStream As New SpFileStream()
                Establezco la velocidad el audio
            pvoz.Volume = pvolumen
                Establezco la velocidad del audio
pvoz.SynchronousSpeakTimeout = pvelocidad
                Guardo la cadena de caracteres en un la unidad C carpeta SEUZ
SpFileStream.Open("C:\SEUZ\" + archivo, SpFileMode, False)
            pvoz.AudioOutputStream = SpFileStream
            pvoz.Speak(info, SpFlags)
            pvoz.WaitUntilDone(Timeout.Infinite)
            SpFileStream.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
            End Sub
End Class
```

Clase usuarios

Clase que me permite manejar los usuarios del sistema

```
Imports MySql.Data.MySqlClient
Clase para cargar usuarios
Public Class usuario
Atributo del usuario
Private unombre As String
    Private uapellido As String
    Private unick As String
    Private uclave As String
    Private uid As Integer
Programa si la conexión falla
    Private bConexionExitosa As Boolean = True
    Creamos un objeto de tipo Connection y configuramos los parámetros de la conexión
Private conexion As New MySqlConnection
Creamos una variable para las consultas
Private consultas As String
    Private comandos As New MySqlCommand

Métodocambiar u obtener la propiedad nombre del usuario
    Public Property Nombre() As String
        ' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return unombre
        End Get
        ' bloque Set para asignar
        ' valor a la provided
        Set(ByVal Value As String)
```

```
            unombre = Value
        End Set
    End Property
```

Método cambiar u obtener la propiedad apellido del usuario

```
Public Property Apellido() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return uapellido
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            uapellido = Value
        End Set
    End Property
```

Metodo cambiar u obtener la propiedad nick del usuario

```
Public Property nick() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return unick
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            unick = Value
        End Set
```

```
    End Property

 Metodo cambiar u obtener la propiedad clave del usuario
Public Property clave() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return uclave
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            uclave = Value
        End Set
    End Property

  Metodo cambiar u obtener la propiedad id del usuario
Public Property id() As Integer
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return uid
        End Get
        ' bloque Set para asignar
' valor a la propiedad
        Set(ByVal Value As Integer)
uid = Value
        End Set
    End Property
```

```
Metodo que me permite buscar un usuario en la base de datos
Public Function buscar(ByRef nick As String, ByRef key As String) As Integer
Dim valor As Integer = 0
        'Try
        'Los parámetros de la sobrecarga con más parámetros son:
        '1. Dirección IP o nombre de la máquina con el servidor de MySQLS
        '2. Nombre de la base de datos
        '3. Nombre de usuario con acceso a la base de datos señalada anteriormente
        '4. Contraseña para el nombre de usuario citado
conexion.ConnectionString = "server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
        conexion.Open()
        'creamos una consulta para la base de datos
consultas = "select * from seuz.usuarios where user='" + nick + "' And clave ='" + key + "'
and estado='A'"
'establecemos la conexion
        comandos.Connection = conexion
        comandos.CommandText = consultas
        'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
        comandos.CommandType = CommandType.Text
        'Creamos un lector de datos (IDataReader) y lo inicializamos
        'con el lector del objeto MySQLCommand
Dim dr As System.Data.IDataReader
dr = comandos.ExecuteReader()
        'Mientras haya datos para leer los mostramos
        While dr.Read()
            'Al igual de los objetos DataRow, la clase IDataReader también tiene
            'un método por defecto .Item que funciona exactamente igual
uid = dr("id")
            unick = dr("user")
```

```
unombre = dr("nombre")
            If unick <> "" Then
                valor = 1
            End If
        End While
        'Cerramos la conexión con el servidor
conexion.Close()
        'Catch ex As Exception
'mostramos los errores si sucedieran
        'MsgBox(ex.ToString, MsgBoxStyle.Information)
'End Try
        Return valor
    End Function

Metodo que me permite Guardar un Nuevo usuario en la base de datos
Public Function grabar() As Integer
        Dim retorno As Integer = 0
        Try
            conexion.ConnectionString =
"server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
            conexion.Open()
            comandos.Connection = conexion
            Dim adap As New MySqlDataAdapter("SELECT *  FROM seuz.usuarios where
nombre='" + unombre + "' and apellido='" + uapellido + "'", conexion)
Dim user As New DataSet
            adap.Fill(user, "usuarios")
            If (user.Tables("usuarios").Rows.Count = 0) Then
                user.Dispose()
                comandos.CommandText = "SELECT max(id)+1 as maximo FROM seuz.usuarios"
```

```
            comandos.CommandType = CommandType.Text
            Dim dr As System.Data.IDataReader
            dr = comandos.ExecuteReader()
            Dim ultimo As Integer
            While dr.Read
               ultimo = dr("maximo")
            End While
            dr.Dispose()
            Dim fecha As String
            fecha = Now().Year.ToString + "-" + Now.Month.ToString + "-" +
Now.Day.ToString + " " + Now.Hour.ToString + ":" + Now.Minute.ToString + ":" +
Now.Second.ToString
comandos.CommandText = "insert into
usuarios(id,user,clave,nombre,apellido,fecha_ingre,estado)values(" + ultimo.ToString + ",'" +
unick + "','" + uclave + "','" & unombre & "','" & uapellido & "','" & fecha & "','A')"
comandos.CommandType = CommandType.Text
            comandos.ExecuteNonQuery()
conexion.Close()
            habla.leer("Informacion Grabada")
Else
            conexion.Close()
            retorno = 1
         End If
         Return retorno
      Catch ex As Exception
         MsgBox(ex.ToString)
      End Try
   End Function

End Class
```

Pantalla temas

Pantalla que me permite seleccionar los diferentes temas que posee el sistema

```
Public Class temas
        Atributo que me permite interactuar con el reconocimiento de voz
Private WithEvents engSR As New System.Speech.Recognition.SpeechRecognizer
        Atributo que me permite interactuar con la sintetización de voz
Private WithEvents engSS As New System.Speech.Synthesis.SpeechSynthesizer

Procedimiento que librera todos los recursos utilizados en esta pantalla
Private Sub temas_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
      Dispose()
      engSR.Dispose()
   End Sub

Procedimiento que carga los comandos que utiliza esta pantalla
Private Sub temas_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
Me.ThemeName = confi.tema
      ' creo los comandos con los que voy a trabajar
Dim itemA As New Speech.Recognition.SrgsGrammar.SrgsItem("windows 7")
      Dim itemB As New Speech.Recognition.SrgsGrammar.SrgsItem("Aqua")
      Dim itemC As New Speech.Recognition.SrgsGrammar.SrgsItem("Breeze")
      Dim itemD As New Speech.Recognition.SrgsGrammar.SrgsItem("High constrast black")
      Dim itemE As New Speech.Recognition.SrgsGrammar.SrgsItem("Office 2007 black")
      Dim itemF As New Speech.Recognition.SrgsGrammar.SrgsItem("Desert")
      Dim itemG As New Speech.Recognition.SrgsGrammar.SrgsItem("Office 2010")
      Dim itemH As New Speech.Recognition.SrgsGrammar.SrgsItem("Office 2007 Silver")
      Dim itemI As New Speech.Recognition.SrgsGrammar.SrgsItem("Vista")
```

```
        Dim itemJ As New Speech.Recognition.SrgsGrammar.SrgsItem("Salir")
' creo la regla con los comandos que tengo q incorporar
Dim rule As New Speech.Recognition.SrgsGrammar.SrgsRule("Comandos", _
         New Speech.Recognition.SrgsGrammar.SrgsOneOf(itemA, itemB, itemC, itemD,
itemE, itemF, itemG, itemH, itemI, itemJ))
' creo el documento de ejemplo
Dim doc As New Speech.Recognition.SrgsGrammar.SrgsDocument
' agrego la regla y la pongo como regla por defecto
doc.Rules.Add(rule)
        doc.Root = rule
' cargo la gramática de la aplicación
engSR.LoadGrammar(New Speech.Recognition.Grammar(doc))
    End Sub

Procedimiento que me permite reconocer los comandos que le dicto al sistema
Private Sub engSR_SpeechRecognized(ByVal sender As Object, ByVal e As
System.Speech.Recognition.SpeechRecognizedEventArgs) Handles
engSR.SpeechRecognized
Verifico que comando dicte
Select Case e.Result.Text
            Case "windows 7"
                confi.temas("Windows 7")
                confi.tema = "Windows 7"
Me.ThemeName = confi.tema
                recarga = 1
                principal.Dispose()
                principal.Show()
            Case "Aqua"
                confi.temas("Aqua")
                confi.tema = "Aqua"
```

```
            Me.ThemeName = confi.tema
            recarga = 1
principal.Dispose()
            principal.Show()
        Case "Breeze"
confi.temas("Breeze")
            confi.tema = "Breeze"
            Me.ThemeName = confi.tema
            recarga = 1
principal.Dispose()
            principal.Show()
        Case "High constrast black"
confi.temas("HighContrastBlack")
            confi.tema = "HighContrastBlack"
            Me.ThemeName = confi.tema
            recarga = 1
            principal.Dispose()
principal.Show()
        Case "Office 2007 black"
            confi.temas("Office2007Black")
            confi.tema = "Office2007Black"
Me.ThemeName = confi.tema
            recarga = 1
principal.Dispose()
            principal.Show()
        Case "Desert"
confi.temas("Desert")
            confi.tema = "Desert"
            Me.ThemeName = confi.tema
            recarga = 1
```

```
            principal.Dispose()
            principal.Show()
        Case "Vista"
            confi.temas("Vista")
            confi.tema = "Vista"
            Me.ThemeName = confi.tema
            recarga = 1
            principal.Dispose()
            principal.Show()
Case "Office 2010"
            confi.temas("Office2010")
confi.tema = "Office2010"
            Me.ThemeName = confi.tema
            recarga = 1
principal.Dispose()
            principal.Show()
        Case "Office 2007 Silver"
confi.temas("Office2007Silver")
            confi.tema = "Office2007Silver"
            Me.ThemeName = confi.tema
            recarga = 1
            principal.Dispose()
principal.Show()
        Case "Salir"
            Me.Close()
    End Select
  End Sub
Procedimiento que carga el tema desert
Private Sub tDesert_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
tDesert.Click
```

```
confi.temas("Desert")
        confi.tema = "Desert"
        Me.ThemeName = confi.tema
        recarga = 1
principal.Dispose()
        principal.Show()
    End Sub
```

Procedimiento que carga el tema vista

```
Private Sub tVista_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
tVista.Click
confi.temas("Vista")
        confi.tema = "Vista"
        Me.ThemeName = confi.tema
        recarga = 1
principal.Dispose()
        principal.Show()
    End Sub
```

Procedimiento que carga el tema Windows 7

```
Private Sub tWindows7_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles tWindows7.Click
        confi.temas("Windows 7")
        confi.tema = "Windows 7"
Me.ThemeName = confi.tema
        recarga = 1
principal.Dispose()
        principal.Show()
    End Sub
```

Procedimiento que carga el tema office 2010

```
Private Sub tOffice2010_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles tOffice2010.Click
        confi.temas("Office2010")
        confi.tema = "Office2010"
Me.ThemeName = confi.tema
        recarga = 1
principal.Dispose()
        principal.Show()
    End Sub
```

Procedimiento que carga el tema breeze

```
Private Sub tBreeze_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
tBreeze.Click
confi.temas("Breeze")
        confi.tema = "Breeze"
        Me.ThemeName = confi.tema
        recarga = 1
principal.Dispose()
        principal.Show()
    End Sub
```

Procedimiento que carga el tema HighContrasBlach

```
Private Sub tHighContrasBlack_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tHighContrasBlack.Click
confi.temas("HighContrastBlack")
        confi.tema = "HighContrastBlack"
        Me.ThemeName = confi.tema
        recarga = 1
        principal.Dispose()
```

```
principal.Show()
   End Sub

Procedimiento que carga el tema Office2007Black
Private Sub tOffice2007Black_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tOffice2007Black.Click
       confi.temas("Office2007Black")
       confi.tema = "Office2007Black"
Me.ThemeName = confi.tema
       recarga = 1
principal.Dispose()
       principal.Show()
   End Sub

Procedimiento que carga el tema Aqua de MAC OS
Private Sub tAqua_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
tAqua.Click
confi.temas("Aqua")
       confi.tema = "Aqua"
       Me.ThemeName = confi.tema
       recarga = 1
principal.Dispose()
       principal.Show()
   End Sub

Procedimiento que carga el tema Office 2007 Silver
Private Sub tOffice2007Silver_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tOffice2007Silver.Click
confi.temas("Office2007Silver")
       confi.tema = "Office2007Silver"
```

```
        Me.ThemeName = confi.tema
        recarga = 1
        principal.Dispose()
principal.Show()
    End Sub
End Class
```

Diseño de la pantalla Temas esto se genera automáticamente cuando se añade un Nuevo componente en el modo Diseño del Visual Studio

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class temas
    Inherits Telerik.WinControls.UI.RadForm
    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
```

```
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim ThemeSource1 As Telerik.WinControls.ThemeSource = New
Telerik.WinControls.ThemeSource()
        Dim CarouselEllipsePath1 As Telerik.WinControls.UI.CarouselEllipsePath = New
Telerik.WinControls.UI.CarouselEllipsePath()
        Me.RadThemeManager1 = New Telerik.WinControls.RadThemeManager()
        Me.Carousel1 = New Telerik.WinControls.UI.RadCarousel()
        Me.tVista = New Telerik.WinControls.UI.RadButtonElement()
        Me.tDesert = New Telerik.WinControls.UI.RadButtonElement()
        Me.tOffice2010 = New Telerik.WinControls.UI.RadButtonElement()
        Me.tWindows7 = New Telerik.WinControls.UI.RadButtonElement()
        Me.tAqua = New Telerik.WinControls.UI.RadButtonElement()
        Me.tBreeze = New Telerik.WinControls.UI.RadButtonElement()
        Me.tHighContrasBlack = New Telerik.WinControls.UI.RadButtonElement()
        Me.tOffice2007Black = New Telerik.WinControls.UI.RadButtonElement()
        Me.tOffice2007Silver = New Telerik.WinControls.UI.RadButtonElement()
        CType(Me.Carousel1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'RadThemeManager1
        '
        ThemeSource1.ThemeLocation = "C:\Program Files\Telerik\RadControls for WinForms
Q2 2010\Examples\Carousel\Carou" & _
            "selSettings\CarouselButtons.xml"
        Me.RadThemeManager1.LoadedThemes.AddRange(New
Telerik.WinControls.ThemeSource() {ThemeSource1})
        '
        'Carousel1
```

```
        '
        Me.Carousel1.AutoLoopPauseCondition =
Telerik.WinControls.UI.AutoLoopPauseConditions.None
        Me.Carousel1.BackColor = System.Drawing.Color.Transparent
        Me.Carousel1.BackgroundImage =
Global.SEUZ.My.Resources.Resources.ZEUS_AQUA_HYDRO_by_ZEUSosX
        Me.Carousel1.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch
        Me.Carousel1.ButtonPositions =
Telerik.WinControls.UI.NavigationButtonsPosition.Top
        CarouselEllipsePath1.Center = New Telerik.WinControls.UI.Point3D(50.0R, 50.0R,
0.0R)
        CarouselEllipsePath1.FinalAngle = -100.0R
        CarouselEllipsePath1.InitialAngle = -90.0R
        CarouselEllipsePath1.U = New Telerik.WinControls.UI.Point3D(-20.0R, -17.0R, -
50.0R)
        CarouselEllipsePath1.V = New Telerik.WinControls.UI.Point3D(30.0R, -25.0R, -60.0R)
        CarouselEllipsePath1.ZScale = 500.0R
        Me.Carousel1.CarouselPath = CarouselEllipsePath1
        Me.Carousel1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.Carousel1.EasingType = Telerik.WinControls.RadEasingType.OutQuart
        Me.Carousel1.EnableAutoLoop = True
        Me.Carousel1.Font = New System.Drawing.Font("Times New Roman", 14.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Carousel1.ForeColor = System.Drawing.Color.White
        Me.Carousel1.Items.AddRange(New Telerik.WinControls.RadItem() {Me.tVista,
Me.tDesert, Me.tOffice2010, Me.tWindows7, Me.tAqua, Me.tBreeze,
Me.tHighContrasBlack, Me.tOffice2007Black, Me.tOffice2007Silver})
        Me.Carousel1.Location = New System.Drawing.Point(0, 0)
        Me.Carousel1.Name = "Carousel1"
```

```
        Me.Carousel1.OpacityChangeCondition =
Telerik.WinControls.UI.OpacityChangeConditions.SelectedIndex
        '
        '
        '
        Me.Carousel1.RootElement.ForeColor = System.Drawing.Color.White
        Me.Carousel1.SelectedIndex = 0
        Me.Carousel1.Size = New System.Drawing.Size(776, 453)
        Me.Carousel1.TabIndex = 0
        Me.Carousel1.Text = "Temas de SEUZ"
        Me.Carousel1.ThemeName = "ControlDefault"
'
        'tVista
        '
        Me.tVista.Name = "tVista"
Me.tVista.Text = "Vista"
        Me.tVista.TextOrientation = System.Windows.Forms.Orientation.Horizontal
        '
        'tDesert
        '
        Me.tDesert.Name = "tDesert"
        Me.tDesert.Text = "Desert"
        '
        'tOffice2010
        '
        Me.tOffice2010.Name = "tOffice2010"
        Me.tOffice2010.Text = "Office 2010"
        '
        'tWindows7
        '
```

```
Me.tWindows7.Name = "tWindows7"
Me.tWindows7.Text = "Windows 7"
'
'tAqua
'
Me.tAqua.Name = "tAqua"
Me.tAqua.Text = "Aqua"
'
'tBreeze
'
Me.tBreeze.Name = "tBreeze"
Me.tBreeze.Text = "Breeze"
'
'tHighContrasBlack
'
Me.tHighContrasBlack.Name = "tHighContrasBlack"
Me.tHighContrasBlack.Text = "HighContrastBlack"
'
'tOffice2007Black
'
Me.tOffice2007Black.Name = "tOffice2007Black"
Me.tOffice2007Black.Text = "Office 2007 Black"
'
'tOffice2007Silver
'
Me.tOffice2007Silver.Image = Nothing
Me.tOffice2007Silver.Name = "tOffice2007Silver"
Me.tOffice2007Silver.Text = "Office 2007 Silver"
'
'temas
```

```
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(776, 453)
        Me.Controls.Add(Me.Carousel1)
        Me.Cursor = System.Windows.Forms.Cursors.Default
Me.Name = "temas"
        '
        '
        '
Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "temas"
        Me.ThemeName = "ControlDefault"
        Me.TopMost = True
        CType(Me.Carousel1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
    End Sub
    Friend WithEvents RadThemeManager1 As Telerik.WinControls.RadThemeManager
    Friend WithEvents Carousel1 As Telerik.WinControls.UI.RadCarousel
    Friend WithEvents tVista As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tDesert As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tOffice2010 As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tWindows7 As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tAqua As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tBreeze As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tHighContrasBlack As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tOffice2007Black As Telerik.WinControls.UI.RadButtonElement
    Friend WithEvents tOffice2007Silver As Telerik.WinControls.UI.RadButtonElement
```

End Class

Clase reglas

Clase que me permite cargar las reglas basadas en el conocimiento que está en la base de datos y las palabras que se utiliza para el reconocimiento de frase

```
'clase para cargar las reglas
Librerías que me permite cargar las reglas gramaticales del sistema
Imports MySql.Data.MySqlClient
Imports SpeechLib
Imports System.Speech.Recognition.SrgsGrammar
Imports System.Speech.Recognition

Public Class reglas
        Atributos que me permiten manipular la base de datos
   Private conexion As New MySqlConnection
Creamos una variable para las consultas
Private consultas As String
   Private comandos As New MySqlCommand
Metodo para cargar la configuración
   Private matriz1, matriz2 As New ArrayList
Private rules As New Speech.Recognition.GrammarBuilder

Metodo que me permite obtener las reglas Gramaticales
Public Property regla() As Speech.Recognition.GrammarBuilder
' bloque Get para devolver
      ' el valor de la propiedad
Get
         Return rules
      End Get
```

```
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As Speech.Recognition.GrammarBuilder)
            rules = Value
        End Set
    End Property
```

Método que me permite crear las reglas gramáticas con el conocimiento de la base de datos

```
Public Sub cargar()
        Try
            ' creo los comandos con los que voy a trabajar
conexion.ConnectionString = "server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
If conexion.State = ConnectionState.Open Then
Else
                conexion.Open()
            End If
            'creamos una consulta para la base de datos
            consultas = "select descripción from seuz.palabras where estado='A'"
            'establecemos la conexion
            comandos.Connection = conexion
comandos.CommandText = consultas
            comandos.CommandType = CommandType.Text
'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
            'Creamos un lector de datos (IDataReader) y lo inicializamos
            'con el lector del objeto MySQLCommand
Dim ejecutar As New Choices
            Dim palabras As System.Data.IDataReader
palabras = comandos.ExecuteReader()
            While palabras.Read
```

```
            ejecutar.Add(palabras("descripcion"))
        End While
        conexion.Close()
        conexion.Open()
        'cargar reglas de conversación
        Dim conversacion As New Choices
        consultas = "select tema from seuz.conocimiento where estado='C'"
        'establecemos la conexion
comandos.Connection = conexion
        comandos.CommandText = consultas
        comandos.CommandType = CommandType.Text
        Dim conver As System.Data.IDataReader
        conver = comandos.ExecuteReader()
        While conver.Read
conversacion.Add(conver("tema"))
        End While
        conexion.Close()
        conexion.Open()
        'comandos.Connection = conexion
        'creamos una consulta para la base de datos
        consultas = "select tema from seuz.conocimiento"
        comandos.CommandText = consultas
        'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
        comandos.CommandType = CommandType.Text
        'Creamos un lector de datos (IDataReader) y lo inicializamos
        'con el lector del objeto MySQLCommand
Dim dr As System.Data.IDataReader
dr = comandos.ExecuteReader()
        'combinamos las palabras con el conocimiento
Dim conocimiento As New Choices
```

```
        While dr.Read
conocimiento.Add(dr("tema"))
        End While
conexion.Close()
        Agrego las palabras clave con el conocimiento del sistema
Dim buscar = New GrammarBuilder("BUSCAR")
buscar.Append(conocimiento)
        Dim cuente = New GrammarBuilder("COMO PASO")
        cuente.Append(conocimiento)
        Dim diga = New GrammarBuilder("CUENTE")
        diga.Append(conocimiento)
        Dim hable = New GrammarBuilder("HABLE")
        hable.Append(conocimiento)
Dim permutations = New Choices()
        permutations.Add(buscar)
        permutations.Add(cuente)
        permutations.Add(diga)
        permutations.Add(hable)
        permutations.Add(ejecutar)
        permutations.Add(conversacion)
        rules.Append(permutations)
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
End Class
```

Pantalla principal

Pantalla principal del sistema

```
Option Explicit On
Librerías que me permite interactuar con la voz, reconocimiento de voz, graficos
Imports System
Imports System.Windows.Forms
Imports System.Drawing
Imports System.Runtime.InteropServices
Imports System.Drawing.Printing
Imports System.IO
Imports SpeechLib
Imports System.Environment
Imports System.Threading
Imports System.Speech.Recognition
Imports System.Speech.Recognition.SrgsGrammar

Public Class principal
    'propiedades para crear un pdf
Private Const AUTHOR As String = "SEUZ, Inc."
    Private Const CREATOR As String = "SEUZs Report"
    Private COPYRIGHT As String = String.Format("Copyright © 2010-{0} by Infragistics,
Inc.", DateTime.Now.Year.ToString())
    Private Const PAGE_NUMBER_TEMPLATE As String = "Page [Page #] of [TotalPages]"
    Private Shared report As Infragistics.Documents.Report.Report = New
Infragistics.Documents.Report.Report()
    Private documentFileNameRoot As String = "Document.pdf"
    Private Shared section As Infragistics.Documents.Report.Section.ISection
        Private Shared exportFileFormat As Infragistics.Documents.Report.FileFormat =
Infragistics.Documents.Report.FileFormat.PDF
    ' Styles
    Private Shared normalStyle As Infragistics.Documents.Report.Text.Style
    Private Shared redSubscriptStyle As Infragistics.Documents.Report.Text.Style
```

```
    Private Shared blueStyle As Infragistics.Documents.Report.Text.Style
    Private Shared redStyle As Infragistics.Documents.Report.Text.Style
    Private Shared boldStyle As Infragistics.Documents.Report.Text.Style
    Private Shared blueUnderlineStyle As Infragistics.Documents.Report.Text.Style
    Private Shared blueLargeStyle As Infragistics.Documents.Report.Text.Style
    Private Shared blackLargeStyle As Infragistics.Documents.Report.Text.Style
    Private Shared greenStyle As Infragistics.Documents.Report.Text.Style

'fin de propiedades para crear pdf
Private opcionesImprPagina As New PageSettings()
    Private cadenaaImpr As String
    Private FuenteImpr As New Font(confi.tipo_letra, 10)
    Private WithEvents engSR As New System.Speech.Recognition.SpeechRecognizer
    Private WithEvents engSS As New System.Speech.Synthesis.SpeechSynthesizer

Objecto utilizado para el reconococimiento de voz
    Private recognizer As SpeechRecognizer
Constructor de la clase principal
#Region "Constructor"
    Public Sub New()
        InitializeComponent()
        SetupStyles()
        SetupDefaultSection()
    End Sub
#End Region

Procedimiento que me permite configurar las propiedades del pdf
    Private Sub SetupReportInfo(ByVal title As String)
        report = New Infragistics.Documents.Report.Report()
```

```
        report.Info.Title = title
        report.Info.Author = AUTHOR
        report.Info.Creator = CREATOR
        report.Info.Copyright = COPYRIGHT
        ' MBS 1/11/08 - BR29211
        report.Preferences.Printing.PaperSize =
Infragistics.Documents.Report.Preferences.Printing.PaperSize.Auto
        report.Preferences.Printing.PaperOrientation =
Infragistics.Documents.Report.Preferences.Printing.PaperOrientation.Auto
        report.Preferences.Printing.FitToMargins = True
        SetupDefaultSection()
    End Sub
```

Metodo que me permite configurar la letra del pdf y el diseño de la pagina

```
    Private Sub SetupDefaultSection()
        section = report.AddSection()
        section.PageSize = Infragistics.Documents.Report.PageSizes.Letter
        section.PageMargins.All = 35
        section.PageNumbering = New Infragistics.Documents.Report.Section.PageNumbering()
        section.PageNumbering.Style = New
Infragistics.Documents.Report.Text.Style(Infragistics.Documents.Graphics.Fonts.Arial,
Infragistics.Documents.Graphics.Brushes.Black)
        section.PageNumbering.Template = PAGE_NUMBER_TEMPLATE
        section.PageNumbering.Continue = True
        section.PageNumbering.SkipFirst = False
        section.PageNumbering.Alignment =
Infragistics.Documents.Report.PageNumberAlignment.Right
        section.PageNumbering.OffsetX = -10
    End Sub
```

Metodo que me permite configurar el estilo del pdf

```
    Private Sub SetupStyles()
        normalStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 12),
Infragistics.Documents.Graphics.Brushes.Black)
        redSubscriptStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 12),
Infragistics.Documents.Graphics.Brushes.Red)
        redSubscriptStyle.FontVariant = Infragistics.Documents.Report.FontVariant.Subscript
        blueStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font(dletras.Font.Name, dletras.Font.Size),
Infragistics.Documents.Graphics.Brushes.Blue)
        redStyle = New
Infragistics.Documents.Report.Text.Style(Infragistics.Documents.Graphics.Fonts.Arial,
Infragistics.Documents.Graphics.Brushes.Red)
        boldStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 12,
Infragistics.Documents.Graphics.FontStyle.Bold),
Infragistics.Documents.Graphics.Brushes.Black)
        blueUnderlineStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 10,
Infragistics.Documents.Graphics.FontStyle.Underline),
Infragistics.Documents.Graphics.Brushes.DarkBlue)

        blueLargeStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 20),
Infragistics.Documents.Graphics.Brushes.DarkBlue)
        blackLargeStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 16),
Infragistics.Documents.Graphics.Brushes.Black)
```

```
        greenStyle = New Infragistics.Documents.Report.Text.Style(New
Infragistics.Documents.Graphics.Font("Arial", 10),
Infragistics.Documents.Graphics.Brushes.DarkGreen)
    End Sub
```

Método que me permite abrir un pdf

```
 Private Shared Sub AddContentToReport(ByVal content As String)
        AddContentToReport(content, blueStyle)
    End Sub
```

Método que me permite agregar información al pdf

```
    Private Shared Sub AddContentToReport(ByVal content As String, ByVal style As
Infragistics.Documents.Report.Text.Style)
        Dim text As Infragistics.Documents.Report.Text.IText = section.AddText()
        text.Style = style
        text.AddContent(content)
    End Sub
 Private Sub PublishReport()
        report.Publish(documentFileNameRoot, exportFileFormat)
    End Sub
```

Metodo que me permite imprimir un informacion determinada

```
Private Sub imprimiendo ()
        Try
            Try
                'Especifica los parámetros de la página actual
                PrintDocument1.DefaultPageSettings = opcionesImprPagina
                'Especifica documento para cuadro de diálogo de impresión y mostrar
                cadenaaImpr = informacion.Text
                PrintDialog1.Document = PrintDocument1
```

```
            Dim result As DialogResult = PrintDialog1.ShowDialog()
            'Si se pulsa Aceptar se imprimirá el documento
If result = DialogResult.OK Then
                PrintDocument1.Print()
            End If
        Catch ex As Exception
'Muestra mensajes de error
            MessageBox.Show(ex.Message)
End Try
    Catch ex As Exception

    End Try
  End Sub

Procedimiento que abre un pdf
Private Sub OpenReport()
    System.Diagnostics.Process.Start(documentFileNameRoot)
  End Sub

Procedimiento que crear un pdf
  Public Sub pdf()
    Try
      SetupReportInfo("Report with Attachments")

      ' Content
      AddContentToReport(informacion.Text)
      ' Attachments
      Dim fileName As String = "C:\Users\falcon\Desktop\SEUZ\SEUZ\fondoA.jpg"
      Dim attachmentName As String = "fondoA.jpg"
      Dim mimeType As String = "fondoA/jpg"
```

```
            Dim description As String = "Seuz 2011"
            report.AttachFile(fileName, attachmentName, mimeType, description)

            PublishReport()
            OpenReport()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

Procedimiento que invoca procedimiento que crear un pdf
Private Sub Tpdf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Tpdf.Click
        pdf()
    End Sub

Procedimiento que llama al procedimiento que imprime informacion
    Private Sub Timprimir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timprimir.Click
        imprimiendo()
    End Sub

Procedimiento que configure la hoja a imprimir
Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
        Dim numCaract As Integer
        Dim numLineas As Integer
        Dim strParaPagina As String
        Dim strFormato As New StringFormat()
'define un rectángulo basándose en la configuración de página
```

```
Dim rectDibujar As New RectangleF( _
        e.MarginBounds.Left, e.MarginBounds.Top, _
        e.MarginBounds.Width, e.MarginBounds.Height)
'Define un área para calcular el texto que cabe en una página
      'Disminuye en una línea la altura para asegurar el texto no se sale
Dim medidaTamaño As New SizeF(e.MarginBounds.Width, _
        e.MarginBounds.Height - FuenteImpr.GetHeight(e.Graphics))
'Cuando escribe cadenas largas, las separa entre palabras
      strFormato.Trimming = StringTrimming.Word
      'Calcula cuantos caracteres y líneas pueden caber basándose en medidaTamaño
e.Graphics.MeasureString(cadenaaImpr, FuenteImpr, _
MedidaTamaño, strFormato, numCaract, numLineas)
      'Calcula las cadenas que caben en una página
      strParaPagina = cadenaaImpr.Substring(0, numCaract)
      'Imprime la cadena contenida en la página actual
e.Graphics.DrawString(strParaPagina, FuenteImpr, _
Brushes.Black, rectDibujar, strFormato)
      'Si hay más texto, indica que hay más páginas
      If numCaract < cadenaaImpr.Length Then
         'Resta el texto que ya se ha imprimido de la cadena
cadenaaImpr = cadenaaImpr.Substring(numCaract)
         e.HasMorePages = True
Else
         e.HasMorePages = False
         'Se ha impreso todo el texto, restaura la cadena
cadenaaImpr = informacion.Text
      End If
   End Sub
```

Procedimiento que librera los recursos utilizados en esta pantalla

```
    Private Sub principal_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
        Dispose()
        End
    End Sub

Procedimiento que carga la configuración y reglas gramaticales
Private Sub principal_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        If (recarga <> 0) Then
            Me.DsAudioPlayer1.Start()
        End If
            Cargo las configuraciones guardadas del sistema
        Me.ThemeName = confi.tema
Me.menuprincipal.ThemeName = confi.tema
        Me.observacion.ThemeName = confi.tema
        Me.fuente.ThemeName = confi.tema
        Me.Tguardar.ThemeName = confi.tema
        Me.Timprimir.ThemeName = confi.tema
        Me.Tpdf.ThemeName = confi.tema
        Me.volumen.ThemeName = confi.tema
        Me.hablado.ThemeName = confi.tema
        Me.informacion.Font = letras
        If recarga = 0 Then
            habla.guardar("bienvenido a seuz " + usuario.Nombre + "  cuáles son sus órdenes ",
"temp.wma")
Dim a As String = "C:\SEUZ\temp.wma"
            DsAudioPlayer1.FileName = a
            DsAudioPlayer1.Start()
        End If
```

```
        Try
Cargo las reglas gramaticales que las cree en el metodo cargar de la clase reglas
            recognizer = New SpeechRecognizer()
            AddHandler recognizer.SpeechRecognized, AddressOf recognizer_SpeechRecognized
            AddHandler recognizer.SpeechRecognitionRejected, AddressOf
recognizer_SpeechRecognitionRejected
            Dim pizzaGrammar = New Grammar(regla.regla)
            pizzaGrammar.Name = "Available programs"
            recognizer.LoadGrammar(pizzaGrammar)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

Procedemiento que me permite verificar si esta funcionando en reconocimiento de voz
Private Sub recognizer_SpeechDetected(ByVal sender As Object, ByVal e As
SpeechDetectedEventArgs)
' MsgBox("e"+e.ToString)
'End Sub

Procedimiento que me permite reconocer frase que no posee el sistema
    Private Sub recognizer_SpeechRecognitionRejected(ByVal sender As Object, ByVal e As
SpeechRecognitionRejectedEventArgs)
        MessageBox.Show("Sorry but the " & e.Result.Text & " phrase could not be
recognized")
    End Sub

Procedimiento que me permite reconocer que comando le dicte al sistema
Private Sub recognizer_SpeechRecognized(ByVal sender As Object, ByVal e As
SpeechRecognizedEventArgs)
```

```
        Try
            Verifico que comando le dicte al sistema
            Select Case e.Result.Text.ToUpper
                Case Is = "TEMAS"
                    Dim temas As New temas
                    temas.Show()
                Case Is = "EXPORTAR A PDF"
                    If (informacion.Text = "") Then
                        DsAudioPlayer1.Close()
habla.guardar("NO hay informacion para guarda", "informacion.wma")
DsAudioPlayer1.FileName = "c:\SEUZ\informacion.wma"
                        DsAudioPlayer1.Start()
                    Else
                        pdf()
                    End If
                Case Is = "IMPRIMIR"
                    If (informacion.Text = "") Then
                        DsAudioPlayer1.Close()
habla.guardar("NO hay informacion para imprimir", "informacion.wma")
DsAudioPlayer1.FileName = "c:\SEUZ\informacion.wma"
                        DsAudioPlayer1.Start()
                    Else
                        imprimiendo()
                    End If
                Case Is = "PARAR"
                    DsAudioPlayer1.Stop()
                Case Is = "APRENDER"
                    Dim aprender As New Aprender
                    If aprender.Created = False Then
                        aprender.Show()
```

```
            End If
        Case Is = "BORRAR"
Me.hablado.Text = ""
        Case Is = "TIPO DE LETRA"
            dletras.ShowDialog()
            informacion.Font = dletras.Font
            confi.establecer_letra(dletras.Font.Size, dletras.Font.Name, dletras.Color.Name)
Case Is = "COLOR"
            color.ShowDialog()
            Me.informacion.ForeColor = color.SelectedColor
        Case Is = "NUEVO USUARIO"
            Dim nuevo_usuario As New ingresar_usuario
nuevo_usuario.Show()
        Case Is = "EXPORTAR A WORD"
            If (informacion.Text = "") Then
                DsAudioPlayer1.Close()
habla.guardar("NO hay informacion para guarda", "informacion.wma")
DsAudioPlayer1.FileName = "c:\SEUZ\informacion.wma"
                DsAudioPlayer1.Start()
            Else
                word()
            End If

        Case Is = "EXPORTAR A WORD 2007"
            If (informacion.Text = "") Then
                DsAudioPlayer1.Close()
habla.guardar("NO hay informacion para guarda", "informacion.wma")
DsAudioPlayer1.FileName = "c:\SEUZ\informacion.wma"
                DsAudioPlayer1.Start()
            Else
```

```
                WORD_2007()
            End If
        Case Is = "EXPORTAR A TXT"
            If (informacion.Text = "") Then
                DsAudioPlayer1.Close()
habla.guardar("NO hay informacion para guarda", "informacion.wma")
DsAudioPlayer1.FileName = "c:\SEUZ\informacion.wma"
                DsAudioPlayer1.Start()
            Else
                ARCHIVO()
            End If
        Case Is = "DE NUEVO"
            DsAudioPlayer1.Start()
        Case Is = "SEUZ PRESENTATE"
DsAudioPlayer1.Close()
            habla.guardar("HOLA YO SOY El SISTEMA DE INTELIGENCIA
ARTIFICIAL QUE POSEE LA TECNOLOGIAS: RECONOCIMIENTO DE VOZ Y
SINTETIZACION DE VOZ SEUZ", "presentate.wma")
DsAudioPlayer1.FileName = "C:\SEUZ\presentate.wma"
            DsAudioPlayer1.Start()
        Case Else
            DsAudioPlayer1.Close()
habla.guardar("Buscando Informacion  ", "ordenes.wma")
DsAudioPlayer1.FileName = "C:\SEUZ\ordenes.wma"
            DsAudioPlayer1.Start()
            infor.buscar(e.Result.Text.ToLower)
            If (infor.observacion.Equals("A")) Then
habla.guardar(infor.contenido, infor.tema + ".wma")
            Else
```

```
                habla.guardar(infor.tema + "Contenido  " + infor.contenido + "Observacion " +
infor.observacion + "Fuente " + infor.fuente, infor.tema + ".wma")
                Me.informacion.Text = infor.tema + infor.contenido
                Me.observacion.Text = infor.observacion
                Me.fuente.Text = infor.fuente
End If
            DsAudioPlayer1.FileName = "C:\SEUZ\" + infor.tema + ".wma"
            DsAudioPlayer1.Start()
        End Select
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
  End Sub

Procedimiento que me permite abrir la pantalla temas
Private Sub menutemas_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menutemas.Click
    Dim temas As New temas
    temas.Show()
  End Sub

Procedimiento que me permite abrir la pantalla Tipo de Letras
  Private Sub Menu_Letras_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Menu_Letras.Click
dletras.ShowDialog()
    informacion.Font = dletras.Font
    confi.establecer_letra(dletras.Font.Size, dletras.Font.Name, dletras.Color.Name)
End Sub

Procedimiento que me permite abrir la pantalla de selección de Color
```

```
    Private Sub Menu_Color_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Menu_Color.Click
        color.ShowDialog()
        Me.informacion.ForeColor = color.SelectedColor
    End Sub

Procedimiento que me permite abrir la pantalla Aprender
    Private Sub menuAprender_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuAprender.Click
        Dim aprender As New Aprender
        aprender.Show()
    End Sub

Procedimiento que me permite crear un archivo de word 2003
Public Sub word()
        Dim word As New Microsoft.Office.Interop.Word.Application
        Dim doc As Microsoft.Office.Interop.Word.Document
        'Try
        doc = word.Documents.Add()
        word.Selection.Font.Name = confi.tipo_letra
        word.Selection.Font.Size = confi.tamaño_letra
        ' word.Selection.Font.Color = confi.color
        Dim insertText As String = informacion.Text
        Dim range As Microsoft.Office.Interop.Word.Range = doc.Range(Start:=0, End:=0)
        range.Text = insertText
        doc.SaveAs("c:\SEUZ\DOCUMENTOS\SEUZ.DOC")
        doc.Close(True)
Process.Start("c:\SEUZ\DOCUMENTOS\SEUZ.DOC")
'Catch ex As Exception
        'MessageBox.Show("Error accessing Word document.")
```

```
        ' End Try
    End Sub

Procedimiento que me permite crear un archivo de word 2007
Public Sub WORD_2007()
        Dim word As New Microsoft.Office.Interop.Word.Application
        Dim doc As Microsoft.Office.Interop.Word.Document
        'Try
        doc = word.Documents.Add()
        word.Selection.Font.Name = confi.tipo_letra
        word.Selection.Font.Size = confi.tamaño_letra
        ' word.Selection.Font.Color = confi.color
        Dim insertText As String = informacion.Text
        Dim range As Microsoft.Office.Interop.Word.Range = doc.Range(Start:=0, End:=0)
        range.Text = insertText
        doc.SaveAs("c:\SEUZ\DOCUMENTOS\SEUZ.DOCX")
        doc.Close(True)
Process.Start("c:\SEUZ\DOCUMENTOS\SEUZ.DOCX")
'Catch ex As Exception
        'MessageBox.Show("Error accessing Word document.")
        'End Try
    End Sub

Procedimiento que me permite crear un archivo txt
    Public Sub ARCHIVO()
Try
‘Crea el archivo
            FileOpen(1, "c:\SEUZ\DOCUMENTOS\SEUZ.TXT", OpenMode.Output)
‘escribe el contenido
PrintLine(1, informacion.Text)
```

```
        FileClose(1)
        Process.Start("c:\SEUZ\DOCUMENTOS\SEUZ.TXT")
      Catch ex As Exception
      End Try
  End Sub
```

Procedimiento que me permite guardar la información del sistema en un format determinado

```
  Private Sub Tguardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Tguardar.Click
Save.Filter = "Archivos de Microsoft Office (*.doc)|*.doc|Archivos de Office 2007-2010
(.*docx)|*.docx|Archivos de Texto (*.txt)|*.txt"
Dim word As New Microsoft.Office.Interop.Word.Application
      Dim doc As Microsoft.Office.Interop.Word.Document
      If Save.ShowDialog() = DialogResult.OK Then
        Select Case Save.FilterIndex
          Case 3
            Try
              ' Crea el archivo
              FileOpen(1, Save.FileName, OpenMode.Output)
' escribe el contenido
              PrintLine(1, informacion.Text)
FileClose(1)
            Catch ex As Exception
              MessageBox.Show("Error accessing Word document.")
            End Try
          Case Else
            Try
              doc = word.Documents.Add()
              word.Selection.Font.Name = confi.tipo_letra
              word.Selection.Font.Size = confi.tamaño_letra
```

```
                Dim insertText As String = informacion.Text
                Dim range As Microsoft.Office.Interop.Word.Range = doc.Range(Start:=0,
End:=0)
                range.Text = insertText
                doc.SaveAs(Save.FileName)
                doc.Close(True)
            Catch ex As Exception
                MessageBox.Show("Error accessing Word document.")
            End Try
        End Select
    End If
  End Sub
```

Procedimiento que me permite abrir la pantalla Ingreso de Usuario Nuevo

```
  Private Sub menu_Ingreso_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menu_Ingreso.Click
Dim nuevo_usuario As New ingresar_usuario
nuevo_usuario.Show()
  End Sub
```

Procedimiento que me permite cambiar el volume de la voz

```
  Private Sub volumen_ValueChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles volumen.ValueChanged
```

Obtengo el valor del volumen y luego lo establezco por medio del metodo volumen

```
      habla.volumen = volumen.Value
```

```
    End Sub

Procedimiento que me permite buscar informacion de forma manual
    Private Sub hablado_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles hablado.KeyPress
        If (Asc(e.KeyChar) = "13") Then
            Select Me.hablado.Text.ToUpper
                Case Is = "Parar"
                    DsAudioPlayer1.Start()
                Case Is = "SEGUIR"
                    DsAudioPlayer1.Start()
                Case Else
                    infor.buscar(Me.hablado.Text)
                    If (infor.observacion.Equals("A")) Then
habla.guardar(infor.contenido, infor.tema + ".wma")
                    Else
                        habla.guardar(infor.tema + "Contenido  " + infor.contenido + "Observacion " +
infor.observacion + "Fuente " + infor.fuente, infor.tema + ".wma")
                        Me.informacion.Text = infor.tema + infor.contenido
                        Me.observacion.Text = infor.observacion
                        Me.fuente.Text = infor.fuente
End If
                    DsAudioPlayer1.FileName = "C:\SEUZ\" + infor.tema + ".wma"
                    DsAudioPlayer1.Start()
            End Select
        End If
    End Sub
    Private Sub hablado_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles hablado.TextChanged
    End Sub
```

```
Private Sub subir_V_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles subir_V.Click
        SendMessageW(Me.Handle, WM_APPCOMMAND, _
  Me.Handle, New IntPtr(APPCOMMAND_VOLUME_UP))
    End Sub

    Private Sub bajar_V_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bajar_V.Click
        SendMessageW(Me.Handle, WM_APPCOMMAND, _
Me.Handle, New IntPtr(APPCOMMAND_VOLUME_DOWN))
    End Sub


End Class
```

Diseño

Este codigo se genera automáticamente cuando estoy diseñando la pantalla

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class principal
    Inherits Telerik.WinControls.UI.RadForm
    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    'NOTE: The following procedure is required by the Windows Form Designer
```

```
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Dim ScopeChannel1 As Mitov.PlotLab.ScopeChannel = New
Mitov.PlotLab.ScopeChannel()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(principal))
        Dim ScopePoints1 As Mitov.PlotLab.ScopePoints = New Mitov.PlotLab.ScopePoints()
        Dim DisplayBrush1 As Mitov.SignalLab.DisplayBrush = New
Mitov.SignalLab.DisplayBrush()
        Dim ScopeChannel2 As Mitov.PlotLab.ScopeChannel = New
Mitov.PlotLab.ScopeChannel()
        Dim ScopePoints2 As Mitov.PlotLab.ScopePoints = New Mitov.PlotLab.ScopePoints()
        Dim DisplayBrush2 As Mitov.SignalLab.DisplayBrush = New
Mitov.SignalLab.DisplayBrush()
        Dim ScopeDataView1 As Mitov.PlotLab.ScopeDataView = New
Mitov.PlotLab.ScopeDataView()
        Dim DisplayViewBackground1 As Mitov.PlotLab.DisplayViewBackground = New
Mitov.PlotLab.DisplayViewBackground()
        Dim DisplayGridLineSettings1 As Mitov.PlotLab.DisplayGridLineSettings = New
Mitov.PlotLab.DisplayGridLineSettings()
        Dim DisplayHighlighting1 As Mitov.PlotLab.DisplayHighlighting = New
Mitov.PlotLab.DisplayHighlighting()
        Dim ChannelItemHighlighting1 As Mitov.PlotLab.ChannelItemHighlighting = New
Mitov.PlotLab.ChannelItemHighlighting()
        Dim ChannelItemHighlighting2 As Mitov.PlotLab.ChannelItemHighlighting = New
Mitov.PlotLab.ChannelItemHighlighting()
```

```
        Dim DisplayItemHighlighting1 As Mitov.PlotLab.DisplayItemHighlighting = New
Mitov.PlotLab.DisplayItemHighlighting()
        Dim DisplayItemHighlighting2 As Mitov.PlotLab.DisplayItemHighlighting = New
Mitov.PlotLab.DisplayItemHighlighting()
        Dim DisplayItemHighlighting3 As Mitov.PlotLab.DisplayItemHighlighting = New
Mitov.PlotLab.DisplayItemHighlighting()
        Dim DisplayItemHighlighting4 As Mitov.PlotLab.DisplayItemHighlighting = New
Mitov.PlotLab.DisplayItemHighlighting()
        Dim ChannelMouseHitPoint1 As Mitov.PlotLab.ChannelMouseHitPoint = New
Mitov.PlotLab.ChannelMouseHitPoint()
        Dim OptionalLabel1 As Mitov.SignalLab.OptionalLabel = New
Mitov.SignalLab.OptionalLabel()
        Dim DisplayItemHighlighting5 As Mitov.PlotLab.DisplayItemHighlighting = New
Mitov.PlotLab.DisplayItemHighlighting()
        Dim PinList1 As OpenWire.Proxy.PinList = New OpenWire.Proxy.PinList()
        Dim ScopeLegendView1 As Mitov.PlotLab.ScopeLegendView = New
Mitov.PlotLab.ScopeLegendView()
        Dim DisplayViewBackground2 As Mitov.PlotLab.DisplayViewBackground = New
Mitov.PlotLab.DisplayViewBackground()
        Dim LegendViewButtonSettings1 As Mitov.PlotLab.LegendViewButtonSettings = New
Mitov.PlotLab.LegendViewButtonSettings()
        Dim LegendGroup1 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel1 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim LegendGroup2 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel2 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
```

```
        Dim LegendGroup3 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel3 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim LegendGroup4 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel4 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim LegendGroup5 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel5 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim LegendGroup6 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel6 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim DisplayViewSize1 As Mitov.PlotLab.DisplayViewSize = New
Mitov.PlotLab.DisplayViewSize()
        Dim LegendGroup7 As Mitov.PlotLab.LegendGroup = New
Mitov.PlotLab.LegendGroup()
        Dim DisplayLabel7 As Mitov.PlotLab.DisplayLabel = New
Mitov.PlotLab.DisplayLabel()
        Dim DisplayTitle1 As Mitov.PlotLab.DisplayTitle = New Mitov.PlotLab.DisplayTitle()
        Dim DisplayViewBackground3 As Mitov.PlotLab.DisplayViewBackground = New
Mitov.PlotLab.DisplayViewBackground()
        Dim DisplayViewSize2 As Mitov.PlotLab.DisplayViewSize = New
Mitov.PlotLab.DisplayViewSize()
        Dim DisplayToolBar1 As Mitov.PlotLab.DisplayToolBar = New
Mitov.PlotLab.DisplayToolBar()
```

```
    Dim DisplayToolBarButtons1 As Mitov.PlotLab.DisplayToolBarButtons = New
Mitov.PlotLab.DisplayToolBarButtons()
    Dim DisplayToolBarButton1 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton2 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton3 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton4 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton5 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton6 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton7 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton8 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton9 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton10 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton11 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim DisplayToolBarButton12 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
    Dim ToolBarPosition1 As Mitov.PlotLab.ToolBarPosition = New
Mitov.PlotLab.ToolBarPosition()
    Dim DisplayTrails1 As Mitov.PlotLab.DisplayTrails = New
Mitov.PlotLab.DisplayTrails()
```

```
        Dim ScopeXAxis1 As Mitov.PlotLab.ScopeXAxis = New Mitov.PlotLab.ScopeXAxis()
        Dim DisplayTitleLabel1 As Mitov.PlotLab.DisplayTitleLabel = New
Mitov.PlotLab.DisplayTitleLabel()
        Dim DisplayViewBackground4 As Mitov.PlotLab.DisplayViewBackground = New
Mitov.PlotLab.DisplayViewBackground()
        Dim ViewButtonSettings1 As Mitov.PlotLab.ViewButtonSettings = New
Mitov.PlotLab.ViewButtonSettings()
        Dim DisplayAxisCursorSettings1 As Mitov.PlotLab.DisplayAxisCursorSettings = New
Mitov.PlotLab.DisplayAxisCursorSettings()
        Dim DisplayGridDirectionalLinesSettings1 As
Mitov.PlotLab.DisplayGridDirectionalLinesSettings = New
Mitov.PlotLab.DisplayGridDirectionalLinesSettings()
        Dim DisplayGridLineSettings2 As Mitov.PlotLab.DisplayGridLineSettings = New
Mitov.PlotLab.DisplayGridLineSettings()
        Dim DisplayGridLineSettings3 As Mitov.PlotLab.DisplayGridLineSettings = New
Mitov.PlotLab.DisplayGridLineSettings()
        Dim AxisFormat1 As Mitov.PlotLab.AxisFormat = New Mitov.PlotLab.AxisFormat()
        Dim MajorTicks1 As Mitov.PlotLab.MajorTicks = New Mitov.PlotLab.MajorTicks()
        Dim TicksStart1 As Mitov.PlotLab.TicksStart = New Mitov.PlotLab.TicksStart()
        Dim MaxScopeAxisValue1 As Mitov.PlotLab.MaxScopeAxisValue = New
Mitov.PlotLab.MaxScopeAxisValue()
        Dim DisplayValueRange1 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue1 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue2 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim BasicAxisValue1 As Mitov.PlotLab.BasicAxisValue = New
Mitov.PlotLab.BasicAxisValue()
```

```
        Dim ScopeMaxSample1 As Mitov.PlotLab.ScopeMaxSample = New
Mitov.PlotLab.ScopeMaxSample()
        Dim ScopeAxisValue1 As Mitov.PlotLab.ScopeAxisValue = New
Mitov.PlotLab.ScopeAxisValue()
        Dim DisplayValueRange2 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue3 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue4 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim BasicAxisValue2 As Mitov.PlotLab.BasicAxisValue = New
Mitov.PlotLab.BasicAxisValue()
        Dim MinorTicks1 As Mitov.PlotLab.MinorTicks = New Mitov.PlotLab.MinorTicks()
        Dim AxisToolBar1 As Mitov.PlotLab.AxisToolBar = New
Mitov.PlotLab.AxisToolBar()
        Dim AxisToolBarButtons1 As Mitov.PlotLab.AxisToolBarButtons = New
Mitov.PlotLab.AxisToolBarButtons()
        Dim DisplayToolBarButton13 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayToolBarButton14 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayToolBarButton15 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayToolBarButton16 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim ScopeXAxisUnitScale1 As Mitov.PlotLab.ScopeXAxisUnitScale = New
Mitov.PlotLab.ScopeXAxisUnitScale()
        Dim DisplayViewSize3 As Mitov.PlotLab.DisplayViewSize = New
Mitov.PlotLab.DisplayViewSize()
```

```
        Dim DisplayAxisZoom1 As Mitov.PlotLab.DisplayAxisZoom = New
Mitov.PlotLab.DisplayAxisZoom()
        Dim DisplayValueRange3 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue5 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue6 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim PinList2 As OpenWire.Proxy.PinList = New OpenWire.Proxy.PinList()
        Dim ScopeYAxis1 As Mitov.PlotLab.ScopeYAxis = New Mitov.PlotLab.ScopeYAxis()
        Dim ScopeAutoScaling1 As Mitov.PlotLab.ScopeAutoScaling = New
Mitov.PlotLab.ScopeAutoScaling()
        Dim ScopeAutoScalingSpace1 As Mitov.PlotLab.ScopeAutoScalingSpace = New
Mitov.PlotLab.ScopeAutoScalingSpace()
        Dim ScopeAutoScalingSpace2 As Mitov.PlotLab.ScopeAutoScalingSpace = New
Mitov.PlotLab.ScopeAutoScalingSpace()
        Dim DisplayTitleLabel2 As Mitov.PlotLab.DisplayTitleLabel = New
Mitov.PlotLab.DisplayTitleLabel()
        Dim DisplayViewBackground5 As Mitov.PlotLab.DisplayViewBackground = New
Mitov.PlotLab.DisplayViewBackground()
        Dim ViewButtonSettings2 As Mitov.PlotLab.ViewButtonSettings = New
Mitov.PlotLab.ViewButtonSettings()
        Dim DisplayAxisCursorSettings2 As Mitov.PlotLab.DisplayAxisCursorSettings = New
Mitov.PlotLab.DisplayAxisCursorSettings()
        Dim DisplayGridDirectionalLinesSettings2 As
Mitov.PlotLab.DisplayGridDirectionalLinesSettings = New
Mitov.PlotLab.DisplayGridDirectionalLinesSettings()
        Dim DisplayGridLineSettings4 As Mitov.PlotLab.DisplayGridLineSettings = New
Mitov.PlotLab.DisplayGridLineSettings()
```

```
        Dim DisplayGridLineSettings5 As Mitov.PlotLab.DisplayGridLineSettings = New
Mitov.PlotLab.DisplayGridLineSettings()
        Dim AxisFormat2 As Mitov.PlotLab.AxisFormat = New Mitov.PlotLab.AxisFormat()
        Dim MajorTicks2 As Mitov.PlotLab.MajorTicks = New Mitov.PlotLab.MajorTicks()
        Dim TicksStart2 As Mitov.PlotLab.TicksStart = New Mitov.PlotLab.TicksStart()
        Dim BasicScopeAxisValue1 As Mitov.PlotLab.BasicScopeAxisValue = New
Mitov.PlotLab.BasicScopeAxisValue()
        Dim DisplayValueRange4 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue7 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue8 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim BasicScopeAxisValue2 As Mitov.PlotLab.BasicScopeAxisValue = New
Mitov.PlotLab.BasicScopeAxisValue()
        Dim DisplayValueRange5 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue9 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue10 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim MinorTicks2 As Mitov.PlotLab.MinorTicks = New Mitov.PlotLab.MinorTicks()
        Dim AxisToolBar2 As Mitov.PlotLab.AxisToolBar = New
Mitov.PlotLab.AxisToolBar()
        Dim AxisToolBarButtons2 As Mitov.PlotLab.AxisToolBarButtons = New
Mitov.PlotLab.AxisToolBarButtons()
        Dim DisplayToolBarButton17 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayToolBarButton18 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
```

```
        Dim DisplayToolBarButton19 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayToolBarButton20 As Mitov.PlotLab.DisplayToolBarButton = New
Mitov.PlotLab.DisplayToolBarButton()
        Dim DisplayViewSize4 As Mitov.PlotLab.DisplayViewSize = New
Mitov.PlotLab.DisplayViewSize()
        Dim DisplayAxisZoom2 As Mitov.PlotLab.DisplayAxisZoom = New
Mitov.PlotLab.DisplayAxisZoom()
        Dim DisplayValueRange6 As Mitov.PlotLab.DisplayValueRange = New
Mitov.PlotLab.DisplayValueRange()
        Dim DisplayOptionalValue11 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayOptionalValue12 As Mitov.PlotLab.DisplayOptionalValue = New
Mitov.PlotLab.DisplayOptionalValue()
        Dim DisplayZoom1 As Mitov.PlotLab.DisplayZoom = New
Mitov.PlotLab.DisplayZoom()
        Dim DsAudioOutputDevice1 As Mitov.AudioLab.DSAudioOutputDevice = New
Mitov.AudioLab.DSAudioOutputDevice()
        Dim SourcePin1 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim StatePin1 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim DsGraph1 As Mitov.AudioLab.DSGraph = New Mitov.AudioLab.DSGraph()
        Dim SourcePin2 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim SinkPin1 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim MasterPumping1 As Mitov.AudioLab.MasterPumping = New
Mitov.AudioLab.MasterPumping()
        Dim SinkPin2 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim StatePin2 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim SinkPin3 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim DsGraph2 As Mitov.AudioLab.DSGraph = New Mitov.AudioLab.DSGraph()
        Dim SourcePin3 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
```

```
        Dim SourcePin4 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim StatePin3 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim StatePin4 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim SourcePin5 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim DsAudioOutputDevice2 As Mitov.AudioLab.DSAudioOutputDevice = New
Mitov.AudioLab.DSAudioOutputDevice()
        Dim SourcePin6 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim StatePin5 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim DsGraph3 As Mitov.AudioLab.DSGraph = New Mitov.AudioLab.DSGraph()
        Dim SourcePin7 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim SinkPin4 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim MasterPumping2 As Mitov.AudioLab.MasterPumping = New
Mitov.AudioLab.MasterPumping()
        Dim SinkPin5 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim StatePin6 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim SinkPin6 As OpenWire.Proxy.SinkPin = New OpenWire.Proxy.SinkPin()
        Dim DsGraph4 As Mitov.AudioLab.DSGraph = New Mitov.AudioLab.DSGraph()
        Dim SourcePin8 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim SourcePin9 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Dim StatePin7 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim StatePin8 As OpenWire.Proxy.StatePin = New OpenWire.Proxy.StatePin()
        Dim SourcePin10 As OpenWire.Proxy.SourcePin = New OpenWire.Proxy.SourcePin()
        Me.Temas = New Telerik.WinControls.UI.RadMenuItem()
        Me.FontTIPO = New Telerik.WinControls.UI.RadMenuItem()
Me.hablado = New Telerik.WinControls.UI.RadTextBox()
        Me.Tguardar = New Telerik.WinControls.UI.RadButton()
        Me.Timprimir = New Telerik.WinControls.UI.RadButton()
Me.Tpdf = New Telerik.WinControls.UI.RadButton()
        Me.dletras = New System.Windows.Forms.FontDialog()
        Me.PrintDialog1 = New System.Windows.Forms.PrintDialog()
```

```
Me.PrintDocument1 = New System.Drawing.Printing.PrintDocument()
Me.Save = New System.Windows.Forms.SaveFileDialog()
Me.RadMenuItem1 = New Telerik.WinControls.UI.RadMenuItem()
Me.u_Ingresar = New Telerik.WinControls.UI.RadMenuItem()
Me.u_modificar = New Telerik.WinControls.UI.RadMenuItem()
Me.ayuda = New Telerik.WinControls.UI.RadMenuItem()
Me.RadLabel1 = New Telerik.WinControls.UI.RadLabel()
Me.volumen = New Telerik.WinControls.UI.RadTrackBar()
Me.RadLabel2 = New Telerik.WinControls.UI.RadLabel()
Me.RadLabel4 = New Telerik.WinControls.UI.RadLabel()
Me.fuente = New Telerik.WinControls.UI.RadTextBox()
Me.RadLabel5 = New Telerik.WinControls.UI.RadLabel()
Me.observacion = New Telerik.WinControls.UI.RadTextBox()
Me.Scope1 = New Mitov.PlotLab.Scope(Me.components)
Me.aprender = New Telerik.WinControls.UI.RadMenuItem()
Me.color = New Telerik.WinControls.RadColorDialog()
Me.menuprincipal = New Telerik.WinControls.UI.RadMenu()
Me.menutemas = New Telerik.WinControls.UI.RadMenuItem()
Me.Menu_Letras = New Telerik.WinControls.UI.RadMenuItem()
Me.Menu_Color = New Telerik.WinControls.UI.RadMenuItem()
Me.RadMenuItem2 = New Telerik.WinControls.UI.RadMenuItem()
Me.menu_Ingreso = New Telerik.WinControls.UI.RadMenuItem()
Me.RadMenuItem4 = New Telerik.WinControls.UI.RadMenuItem()
Me.menuAprender = New Telerik.WinControls.UI.RadMenuItem()
Me.informacion = New System.Windows.Forms.RichTextBox()
Me.DsAudioOut1 = New Mitov.AudioLab.DSAudioOut(Me.components)
Me.DsAudioPlayer1 = New Mitov.AudioLab.DSAudioPlayer(Me.components)
Me.DsAudioOut2 = New Mitov.AudioLab.DSAudioOut(Me.components)
Me.DsAudioPlayer2 = New Mitov.AudioLab.DSAudioPlayer(Me.components)
Me.PictureBox1 = New System.Windows.Forms.PictureBox()
```

```
        CType(Me.hablado, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.Tguardar, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.Timprimir, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.Tpdf, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.volumen, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.fuente, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.observacion, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.Scope1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.menuprincipal, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.DsAudioOut1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.DsAudioPlayer1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.DsAudioOut2, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.DsAudioPlayer2, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'Temas
        '
        Me.Temas.Class = ""
    Me.Temas.Name = "Temas"
        Me.Temas.Text = "Temas"
        '
        'FontTIPO
        '
        Me.FontTIPO.Class = ""
```

```
        Me.FontTIPO.Name = "FontTIPO"
        Me.FontTIPO.Text = "Tipo de Letra"
'
        'hablado
        '
        Me.hablado.Location = New System.Drawing.Point(45, 760)
Me.hablado.Multiline = True
        Me.hablado.Name = "hablado"
'
        '
        '
        Me.hablado.RootElement.StretchVertically = True
        Me.hablado.Size = New System.Drawing.Size(383, 36)
Me.hablado.TabIndex = 16
        Me.hablado.TabStop = False
'
        'Tguardar
        '
        Me.Tguardar.Location = New System.Drawing.Point(754, 697)
        Me.Tguardar.Name = "Tguardar"
        Me.Tguardar.Size = New System.Drawing.Size(124, 41)
Me.Tguardar.TabIndex = 18
        Me.Tguardar.Text = "Guardar"
'
        'Timprimir
        '
        Me.Timprimir.Location = New System.Drawing.Point(1056, 697)
        Me.Timprimir.Name = "Timprimir"
        Me.Timprimir.Size = New System.Drawing.Size(128, 40)
        Me.Timprimir.TabIndex = 20
```

```
        Me.Timprimir.Text = "Imprimir"
        '
        'Tpdf
        '
        Me.Tpdf.Location = New System.Drawing.Point(910, 697)
        Me.Tpdf.Name = "Tpdf"
        Me.Tpdf.Size = New System.Drawing.Size(124, 41)
        Me.Tpdf.TabIndex = 19
        Me.Tpdf.Text = "Exportar a Pdf"
'
        'PrintDialog1
        '
        Me.PrintDialog1.UseEXDialog = True
        '
        'PrintDocument1
'
        '
        'RadMenuItem1
        '
        Me.RadMenuItem1.Items.AddRange(New Telerik.WinControls.RadItem()
{Me.u_Ingresar, Me.u_modificar})
        Me.RadMenuItem1.Name = "RadMenuItem1"
        Me.RadMenuItem1.Text = "Usuario"
'
        'u_Ingresar
        '
        Me.u_Ingresar.Name = "u_Ingresar"
        Me.u_Ingresar.Text = "Ingresar"
        '
        'u_modificar
```

```
        '
        Me.u_modificar.Name = "u_modificar"
        Me.u_modificar.Text = "Moficar"
        '
        'ayuda
        '
        Me.ayuda.Name = "ayuda"
        Me.ayuda.Text = "Ayuda"
        '
        'RadLabel1
        '
        Me.RadLabel1.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.RadLabel1.Location = New System.Drawing.Point(745, 43)
        Me.RadLabel1.Name = "RadLabel1"
        Me.RadLabel1.Size = New System.Drawing.Size(83, 22)
        Me.RadLabel1.TabIndex = 22
        Me.RadLabel1.Text = "Contenido"
        Me.RadLabel1.ThemeName = "Desert"
        '
        'volumen
        '
        Me.volumen.Location = New System.Drawing.Point(45, 709)
        Me.volumen.Maximum = 100
        Me.volumen.Minimum = 1
        Me.volumen.Name = "volumen"
        Me.volumen.Size = New System.Drawing.Size(374, 31)
        Me.volumen.SliderAreaColor1 = System.Drawing.Color.FromArgb(CType(CType(140,
Byte), Integer), CType(CType(137, Byte), Integer), CType(CType(118, Byte), Integer))
```

```
        Me.volumen.SliderAreaColor2 = System.Drawing.Color.FromArgb(CType(CType(140,
Byte), Integer), CType(CType(137, Byte), Integer), CType(CType(118, Byte), Integer))
        Me.volumen.TabIndex = 23
        Me.volumen.Text = "RadTrackBar1"
        Me.volumen.TicksColor = System.Drawing.Color.FromArgb(CType(CType(140, Byte),
Integer), CType(CType(137, Byte), Integer), CType(CType(118, Byte), Integer))
        Me.volumen.TickStyle = Telerik.WinControls.Enumerations.TickStyles.BottomRight
        Me.volumen.Value = 50
        '
        'RadLabel2
        '
        Me.RadLabel2.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.RadLabel2.Location = New System.Drawing.Point(45, 673)
        Me.RadLabel2.Name = "RadLabel2"
        Me.RadLabel2.Size = New System.Drawing.Size(73, 22)
        Me.RadLabel2.TabIndex = 24
        Me.RadLabel2.Text = "Volumen"
        Me.RadLabel2.ThemeName = "Desert"
        '
        'RadLabel4
        '
        Me.RadLabel4.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.RadLabel4.Location = New System.Drawing.Point(754, 500)
        Me.RadLabel4.Name = "RadLabel4"
        Me.RadLabel4.Size = New System.Drawing.Size(118, 22)
Me.RadLabel4.TabIndex = 27
        Me.RadLabel4.Text = "Observaciones"
Me.RadLabel4.ThemeName = "Desert"
```

```
        '
        'fuente
        '
        Me.fuente.Enabled = False
        Me.fuente.Location = New System.Drawing.Point(751, 628)
        Me.fuente.Multiline = True
        Me.fuente.Name = "fuente"
        '
        '
        '
        Me.fuente.RootElement.StretchVertically = True
        Me.fuente.Size = New System.Drawing.Size(426, 45)
        Me.fuente.TabIndex = 29
        Me.fuente.TabStop = False
        '
        'RadLabel5
        '
        Me.RadLabel5.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.RadLabel5.Location = New System.Drawing.Point(754, 590)
        Me.RadLabel5.Name = "RadLabel5"
        Me.RadLabel5.Size = New System.Drawing.Size(59, 22)
        Me.RadLabel5.TabIndex = 30
        Me.RadLabel5.Text = "Fuente"
        Me.RadLabel5.ThemeName = "Desert"
        '
        'observacion
        '
        Me.observacion.Enabled = False
        Me.observacion.Location = New System.Drawing.Point(749, 530)
```

```
Me.observacion.Multiline = True
        Me.observacion.Name = "observacion"
'
        '
        '
        Me.observacion.RootElement.StretchVertically = True
        Me.observacion.Size = New System.Drawing.Size(426, 43)
Me.observacion.TabIndex = 34
        Me.observacion.TabStop = False
        '
        'Scope1
        '
        Me.Scope1.BackColor = System.Drawing.Color.FromArgb(CType(CType(192, Byte),
Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
ScopeChannel1.Color = System.Drawing.Color.FromArgb(CType(CType(192, Byte),
Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        ScopeChannel1.InternalData =
CType(resources.GetObject("ScopeChannel1.InternalData"), Vcl.VclBinaryData)
        ScopeChannel1.Name = "Channel0"
        DisplayBrush1.Color = System.Drawing.Color.FromArgb(CType(CType(255, Byte),
Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayBrush1.Style = Vcl.BrushStyle.Solid
        ScopePoints1.Brush = DisplayBrush1
        ScopePoints1.InternalData = CType(resources.GetObject("ScopePoints1.InternalData"),
Vcl.VclBinaryData)
        ScopePoints1.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        ScopePoints1.Size = CType(3UI, UInteger)
```

```
        ScopeChannel1.Points = ScopePoints1
        ScopeChannel1.Tag = 0
        ScopeChannel1.XAxisIndex = 0
        ScopeChannel1.YAxisIndex = 0
        ScopeChannel2.Color = System.Drawing.Color.FromArgb(CType(CType(192, Byte),
Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        ScopeChannel2.InternalData =
CType(resources.GetObject("ScopeChannel2.InternalData"), Vcl.VclBinaryData)
        ScopeChannel2.Name = "Channel1"
        DisplayBrush2.Color = System.Drawing.Color.FromArgb(CType(CType(255, Byte),
Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayBrush2.Style = Vcl.BrushStyle.Solid
        ScopePoints2.Brush = DisplayBrush2
        ScopePoints2.InternalData = CType(resources.GetObject("ScopePoints2.InternalData"),
Vcl.VclBinaryData)
        ScopePoints2.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        ScopePoints2.Size = CType(3UI, UInteger)
        ScopeChannel2.Points = ScopePoints2
        ScopeChannel2.Tag = 0
        ScopeChannel2.XAxisIndex = 0
        ScopeChannel2.YAxisIndex = 0
        Me.Scope1.Channels.AddRange(New Mitov.PlotLab.ScopeChannel() {ScopeChannel1,
ScopeChannel2})
        DisplayViewBackground1.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeDataView1.Background = DisplayViewBackground1
```

```
        DisplayGridLineSettings1.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer),
CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        ScopeDataView1.Border = DisplayGridLineSettings1
        ScopeDataView1.InternalData =
CType(resources.GetObject("ScopeDataView1.InternalData"), Vcl.VclBinaryData)
        Me.Scope1.DataView = ScopeDataView1
        ChannelItemHighlighting1.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.ChannelLinks = ChannelItemHighlighting1
        ChannelItemHighlighting2.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.Channels = ChannelItemHighlighting2
        DisplayItemHighlighting1.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.CursorLinks = DisplayItemHighlighting1
        DisplayItemHighlighting2.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.Cursors = DisplayItemHighlighting2
        DisplayItemHighlighting3.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.Ellipses = DisplayItemHighlighting3
        DisplayItemHighlighting4.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.Markers = DisplayItemHighlighting4
        ChannelMouseHitPoint1.Color = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        OptionalLabel1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
```

```
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 11, New Vcl.FontStyles(0))
        OptionalLabel1.Text = ""
        ChannelMouseHitPoint1.PointLabel = OptionalLabel1
        DisplayHighlighting1.MouseHitPoint = ChannelMouseHitPoint1
        DisplayItemHighlighting5.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayHighlighting1.Zones = DisplayItemHighlighting5
        Me.Scope1.Highlighting = DisplayHighlighting1
        PinList1.ConnectionData = CType(resources.GetObject("PinList1.ConnectionData"),
OpenWire.PinConnections)
        Me.Scope1.InputPins = PinList1
        Me.Scope1.InternalData = CType(resources.GetObject("Scope1.InternalData"),
Vcl.VclBinaryData)
        ScopeLegendView1.Align = Mitov.PlotLab.ViewAlign.Right
        DisplayViewBackground2.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeLegendView1.Background = DisplayViewBackground2
        LegendViewButtonSettings1.BorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.ButtonColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(0, Byte),
Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.DisabledColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.GlyphColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
```

```
        LegendViewButtonSettings1.MouseDownBorderColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.MouseDownColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.MouseOverBorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        LegendViewButtonSettings1.MouseOverColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeLegendView1.Buttons = LegendViewButtonSettings1
        DisplayLabel1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel1.Text = ""
        LegendGroup1.Caption = DisplayLabel1
        ScopeLegendView1.ChannelLinks = LegendGroup1
        DisplayLabel2.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel2.Text = ""
        LegendGroup2.Caption = DisplayLabel2
        ScopeLegendView1.Channels = LegendGroup2
        DisplayLabel3.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
```

```
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel3.Text = ""
        LegendGroup3.Caption = DisplayLabel3
        ScopeLegendView1.CursorLinks = LegendGroup3
        DisplayLabel4.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel4.Text = ""
        LegendGroup4.Caption = DisplayLabel4
        ScopeLegendView1.Cursors = LegendGroup4
        DisplayLabel5.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel5.Text = ""
        LegendGroup5.Caption = DisplayLabel5
        ScopeLegendView1.Ellipses = LegendGroup5
        ScopeLegendView1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 11, New Vcl.FontStyles(0))
        ScopeLegendView1.InternalData =
CType(resources.GetObject("ScopeLegendView1.InternalData"), Vcl.VclBinaryData)
        DisplayLabel6.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel6.Text = ""
```

```
        LegendGroup6.Caption = DisplayLabel6
        ScopeLegendView1.MarkerGroups = LegendGroup6
        DisplayViewSize1.Size = CType(100UI, UInteger)
        ScopeLegendView1.ViewSize = DisplayViewSize1
        DisplayLabel7.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(0))
        DisplayLabel7.Text = ""
        LegendGroup7.Caption = DisplayLabel7
        ScopeLegendView1.Zones = LegendGroup7
        Me.Scope1.Legend = ScopeLegendView1
        Me.Scope1.Location = New System.Drawing.Point(32, 496)
        Me.Scope1.Name = "Scope1"
        Me.Scope1.RefreshInterval = CType(100UI, UInteger)
        Me.Scope1.Size = New System.Drawing.Size(397, 177)
        Me.Scope1.SizeLimit = CType(0UI, UInteger)
        Me.Scope1.TabIndex = 35
        DisplayViewBackground3.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayTitle1.Background = DisplayViewBackground3
        DisplayTitle1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 21, New Vcl.FontStyles(1))
        DisplayTitle1.Text = ""
        DisplayViewSize2.Size = CType(100UI, UInteger)
        DisplayTitle1.ViewSize = DisplayViewSize2
        DisplayTitle1.Visible = False
        Me.Scope1.Title = DisplayTitle1
```

DisplayToolBar1.BorderColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBar1.ButtonColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBarButtons1.Copy = DisplayToolBarButton1

DisplayToolBarButtons1.Hold = DisplayToolBarButton2

DisplayToolBarButtons1.Print = DisplayToolBarButton3

DisplayToolBarButtons1.Save = DisplayToolBarButton4

DisplayToolBarButtons1.Setup = DisplayToolBarButton5

DisplayToolBarButtons1.XYZoomOn = DisplayToolBarButton6

DisplayToolBarButtons1.ZoomIn = DisplayToolBarButton7

DisplayToolBarButtons1.ZoomMode = DisplayToolBarButton8

DisplayToolBarButtons1.ZoomNext = DisplayToolBarButton9

DisplayToolBarButtons1.ZoomOff = DisplayToolBarButton10

DisplayToolBarButtons1.ZoomOut = DisplayToolBarButton11

DisplayToolBarButtons1.ZoomPrevious = DisplayToolBarButton12

DisplayToolBar1.Buttons = DisplayToolBarButtons1

DisplayToolBar1.DisabledColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBar1.GlyphColor = System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBar1.MouseDownBorderColor = System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBar1.MouseDownButtonColor = System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))

DisplayToolBar1.MouseOverBorderColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))

```
        DisplayToolBar1.MouseOverButtonColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayToolBar1.Position = ToolBarPosition1
        DisplayToolBar1.Visible = False
        Me.Scope1.ToolBar = DisplayToolBar1
        DisplayTrails1.Color = System.Drawing.Color.FromArgb(CType(CType(255, Byte),
Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        DisplayTrails1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 11, New Vcl.FontStyles(0))
        Me.Scope1.Trails = DisplayTrails1
        Me.Scope1.Visible = True
        ScopeXAxis1.Align = Mitov.PlotLab.ViewAlign.Bottom
        DisplayTitleLabel1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(1))
        DisplayTitleLabel1.Text = ""
        ScopeXAxis1.AxisLabel = DisplayTitleLabel1
        DisplayViewBackground4.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeXAxis1.Background = DisplayViewBackground4
        ViewButtonSettings1.MouseDownBorderColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        ViewButtonSettings1.MouseDownColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
```

```
        ViewButtonSettings1.MouseOverBorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        ViewButtonSettings1.MouseOverColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeXAxis1.Button = ViewButtonSettings1
        ScopeXAxis1.Color = System.Drawing.Color.FromArgb(CType(CType(255, Byte),
Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        ScopeXAxis1.CursorSettings = DisplayAxisCursorSettings1
        DisplayGridLineSettings2.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer),
CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        DisplayGridDirectionalLinesSettings1.Lines = DisplayGridLineSettings2
        DisplayGridLineSettings3.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        DisplayGridDirectionalLinesSettings1.ZeroLine = DisplayGridLineSettings3
        ScopeXAxis1.DataView = DisplayGridDirectionalLinesSettings1
        ScopeXAxis1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 11, New Vcl.FontStyles(0))
        AxisFormat1.Precision = 3
        ScopeXAxis1.Format = AxisFormat1
        ScopeXAxis1.InternalData = CType(resources.GetObject("ScopeXAxis1.InternalData"),
Vcl.VclBinaryData)
        TicksStart1.StartFrom = 0.0R
```

```
        MajorTicks1.StartFrom = TicksStart1
        MajorTicks1.Step = 10.0R
        ScopeXAxis1.MajorTicks = MajorTicks1
        MaxScopeAxisValue1.DataValue = 1024.0R
        MaxScopeAxisValue1.InternalData =
CType(resources.GetObject("MaxScopeAxisValue1.InternalData"), Vcl.VclBinaryData)
        DisplayOptionalValue1.Enabled = False
        DisplayOptionalValue1.Value = 1000.0R
        DisplayValueRange1.High = DisplayOptionalValue1
        DisplayOptionalValue2.Enabled = False
        DisplayOptionalValue2.Value = -1000.0R
        DisplayValueRange1.Low = DisplayOptionalValue2
        MaxScopeAxisValue1.Range = DisplayValueRange1
        BasicAxisValue1.InternalData =
CType(resources.GetObject("BasicAxisValue1.InternalData"), Vcl.VclBinaryData)
        BasicAxisValue1.Value = 1024.0R
        MaxScopeAxisValue1.Tick = BasicAxisValue1
        MaxScopeAxisValue1.Value = 1024.0R
        ScopeXAxis1.Max = MaxScopeAxisValue1
        ScopeMaxSample1.Value = CType(1024UI, UInteger)
        ScopeXAxis1.MaxSample = ScopeMaxSample1
        ScopeAxisValue1.DataValue = 0.0R
        ScopeAxisValue1.InternalData =
CType(resources.GetObject("ScopeAxisValue1.InternalData"), Vcl.VclBinaryData)
        DisplayOptionalValue3.Enabled = False
        DisplayOptionalValue3.Value = 1000.0R
        DisplayValueRange2.High = DisplayOptionalValue3
        DisplayOptionalValue4.Enabled = False
        DisplayOptionalValue4.Value = -1000.0R
        DisplayValueRange2.Low = DisplayOptionalValue4
```

```
        ScopeAxisValue1.Range = DisplayValueRange2
        BasicAxisValue2.InternalData =
CType(resources.GetObject("BasicAxisValue2.InternalData"), Vcl.VclBinaryData)
        BasicAxisValue2.Value = 0.0R
        ScopeAxisValue1.Tick = BasicAxisValue2
        ScopeAxisValue1.Value = 0.0R
        ScopeXAxis1.Min = ScopeAxisValue1
        MinorTicks1.Count = 0
        ScopeXAxis1.MinorTicks = MinorTicks1
        AxisToolBar1.BorderColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte),
Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar1.ButtonColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte),
Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBarButtons1.ZoomIn = DisplayToolBarButton13
        AxisToolBarButtons1.ZoomOff = DisplayToolBarButton14
        AxisToolBarButtons1.ZoomOneDir = DisplayToolBarButton15
        AxisToolBarButtons1.ZoomOut = DisplayToolBarButton16
        AxisToolBar1.Buttons = AxisToolBarButtons1
        AxisToolBar1.DisabledColor = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar1.GlyphColor = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar1.MouseDownBorderColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar1.MouseDownButtonColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
```

```
        AxisToolBar1.MouseOverBorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar1.MouseOverButtonColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeXAxis1.ToolBar = AxisToolBar1
        ScopeXAxis1.TrackColor = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeXAxisUnitScale1.Exponent = 0
        ScopeXAxis1.UnitScale = ScopeXAxisUnitScale1
        DisplayViewSize3.Size = CType(100UI, UInteger)
        ScopeXAxis1.ViewSize = DisplayViewSize3
        ScopeXAxis1.Visible = False
        DisplayOptionalValue5.Enabled = False
        DisplayOptionalValue5.Value = 100000000.0R
        DisplayValueRange3.High = DisplayOptionalValue5
        DisplayOptionalValue6.Enabled = True
        DisplayOptionalValue6.Value = 0.0000001R
        DisplayValueRange3.Low = DisplayOptionalValue6
        DisplayAxisZoom1.Range = DisplayValueRange3
        ScopeXAxis1.Zooming = DisplayAxisZoom1
        Me.Scope1.XAxis = ScopeXAxis1
        PinList2.ConnectionData = CType(resources.GetObject("PinList2.ConnectionData"),
OpenWire.PinConnections)
        Me.Scope1.XInputPins = PinList2
        ScopeYAxis1.Align = Mitov.PlotLab.ViewAlign.Left
        ScopeAutoScaling1.MaxHistory = CType(20UI, UInteger)
        ScopeAutoScalingSpace1.Space = CType(20UI, UInteger)
        ScopeAutoScalingSpace1.Threshold = CType(10UI, UInteger)
```

```
        ScopeAutoScaling1.SpaceAbove = ScopeAutoScalingSpace1
        ScopeAutoScalingSpace2.Space = CType(20UI, UInteger)
        ScopeAutoScalingSpace2.Threshold = CType(10UI, UInteger)
        ScopeAutoScaling1.SpaceBelow = ScopeAutoScalingSpace2
        ScopeYAxis1.AutoScaling = ScopeAutoScaling1
        DisplayTitleLabel2.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 13, New Vcl.FontStyles(1))
        DisplayTitleLabel2.Text = "Y Axis"
        ScopeYAxis1.AxisLabel = DisplayTitleLabel2
        DisplayViewBackground5.Color = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeYAxis1.Background = DisplayViewBackground5
        ViewButtonSettings2.MouseDownBorderColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        ViewButtonSettings2.MouseDownColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ViewButtonSettings2.MouseOverBorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        ViewButtonSettings2.MouseOverColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeYAxis1.Button = ViewButtonSettings2
        ScopeYAxis1.Color = System.Drawing.Color.FromArgb(CType(CType(255, Byte),
Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        ScopeYAxis1.CursorSettings = DisplayAxisCursorSettings2
```

```
        DisplayGridLineSettings4.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer),
CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        DisplayGridDirectionalLinesSettings2.Lines = DisplayGridLineSettings4
        DisplayGridLineSettings5.Pen = New
Vcl.VclPen(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), Vcl.PenMode.Copy,
Vcl.PenStyle.Solid, 1)
        DisplayGridDirectionalLinesSettings2.ZeroLine = DisplayGridLineSettings5
        ScopeYAxis1.DataView = DisplayGridDirectionalLinesSettings2
        ScopeYAxis1.Font = New
Vcl.VclFont(System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer)), "Arial",
Vcl.FontCharset.[Default], Vcl.FontPitch.[Default], True, 11, New Vcl.FontStyles(0))
        AxisFormat2.Precision = 3
        ScopeYAxis1.Format = AxisFormat2
        ScopeYAxis1.InternalData = CType(resources.GetObject("ScopeYAxis1.InternalData"),
Vcl.VclBinaryData)
        TicksStart2.StartFrom = 0.0R
        MajorTicks2.StartFrom = TicksStart2
        MajorTicks2.Step = 10.0R
        ScopeYAxis1.MajorTicks = MajorTicks2
        BasicScopeAxisValue1.DataValue = 1000.0R
        BasicScopeAxisValue1.InternalData =
CType(resources.GetObject("BasicScopeAxisValue1.InternalData"), Vcl.VclBinaryData)
        DisplayOptionalValue7.Enabled = False
        DisplayOptionalValue7.Value = 1000.0R
        DisplayValueRange4.High = DisplayOptionalValue7
        DisplayOptionalValue8.Enabled = False
```

```
        DisplayOptionalValue8.Value = -1000.0R
        DisplayValueRange4.Low = DisplayOptionalValue8
        BasicScopeAxisValue1.Range = DisplayValueRange4
        BasicScopeAxisValue1.Value = 1000.0R
        ScopeYAxis1.Max = BasicScopeAxisValue1
        BasicScopeAxisValue2.DataValue = -1000.0R
        BasicScopeAxisValue2.InternalData =
CType(resources.GetObject("BasicScopeAxisValue2.InternalData"), Vcl.VclBinaryData)
        DisplayOptionalValue9.Enabled = False
        DisplayOptionalValue9.Value = 1000.0R
        DisplayValueRange5.High = DisplayOptionalValue9
        DisplayOptionalValue10.Enabled = False
        DisplayOptionalValue10.Value = -1000.0R
        DisplayValueRange5.Low = DisplayOptionalValue10
        BasicScopeAxisValue2.Range = DisplayValueRange5
        BasicScopeAxisValue2.Value = -1000.0R
        ScopeYAxis1.Min = BasicScopeAxisValue2
        MinorTicks2.Count = 0
        ScopeYAxis1.MinorTicks = MinorTicks2
        AxisToolBar2.BorderColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte),
Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar2.ButtonColor = System.Drawing.Color.FromArgb(CType(CType(0, Byte),
Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBarButtons2.ZoomIn = DisplayToolBarButton17
        AxisToolBarButtons2.ZoomOff = DisplayToolBarButton18
        AxisToolBarButtons2.ZoomOneDir = DisplayToolBarButton19
        AxisToolBarButtons2.ZoomOut = DisplayToolBarButton20
        AxisToolBar2.Buttons = AxisToolBarButtons2
        AxisToolBar2.DisabledColor = System.Drawing.Color.FromArgb(CType(CType(0,
Byte), Integer), CType(CType(128, Byte), Integer), CType(CType(0, Byte), Integer))
```

```
        AxisToolBar2.GlyphColor = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar2.MouseDownBorderColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar2.MouseDownButtonColor =
System.Drawing.Color.FromArgb(CType(CType(128, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar2.MouseOverBorderColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer))
        AxisToolBar2.MouseOverButtonColor =
System.Drawing.Color.FromArgb(CType(CType(0, Byte), Integer), CType(CType(128,
Byte), Integer), CType(CType(0, Byte), Integer))
        ScopeYAxis1.ToolBar = AxisToolBar2
        ScopeYAxis1.TrackColor = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(0, Byte), Integer))
        DisplayViewSize4.Size = CType(100UI, UInteger)
        ScopeYAxis1.ViewSize = DisplayViewSize4
        ScopeYAxis1.Visible = False
        DisplayOptionalValue11.Enabled = False
        DisplayOptionalValue11.Value = 100000000.0R
        DisplayValueRange6.High = DisplayOptionalValue11
        DisplayOptionalValue12.Enabled = True
        DisplayOptionalValue12.Value = 0.0000001R
        DisplayValueRange6.Low = DisplayOptionalValue12
        DisplayAxisZoom2.Range = DisplayValueRange6
        ScopeYAxis1.Zooming = DisplayAxisZoom2
        Me.Scope1.YAxis = ScopeYAxis1
```

```
        DisplayZoom1.SelectionColor = System.Drawing.Color.FromArgb(CType(CType(255,
Byte), Integer), CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.Scope1.Zooming = DisplayZoom1
        '
        'aprender
'
        Me.aprender.Class = ""
        Me.aprender.Name = "aprender"
        Me.aprender.Text = "Aprender"
        '
        'color
'
        Me.color.SelectedColor = System.Drawing.Color.Red
        Me.color.SelectedHslColor = Telerik.WinControls.HslColor.FromAhsl(0.0R, 1.0R,
1.0R)
        '
        'menuprincipal
        '
        Me.menuprincipal.Items.AddRange(New Telerik.WinControls.RadItem()
{Me.menutemas, Me.Menu_Letras, Me.Menu_Color, Me.RadMenuItem2,
Me.menuAprender})
        Me.menuprincipal.Location = New System.Drawing.Point(0, 0)
        Me.menuprincipal.Name = "menuprincipal"
        Me.menuprincipal.Size = New System.Drawing.Size(1189, 24)
        Me.menuprincipal.TabIndex = 36
        Me.menuprincipal.Text = "RadMenu1"
        '
        'menutemas
        '
        Me.menutemas.Name = "menutemas"
```

```
Me.menutemas.Text = "Temas"
        '
        'Menu_Letras
        '
        Me.Menu_Letras.Name = "Menu_Letras"
        Me.Menu_Letras.Text = "Tipo Letra"
        '
        'Menu_Color
        '
        Me.Menu_Color.Name = "Menu_Color"
Me.Menu_Color.Text = "Color"
        '
        'RadMenuItem2
        '
        Me.RadMenuItem2.Items.AddRange(New Telerik.WinControls.RadItem()
{Me.menu_Ingreso, Me.RadMenuItem4})
        Me.RadMenuItem2.Name = "RadMenuItem2"
        Me.RadMenuItem2.Text = "Usuarios"
'
        'menu_Ingreso
        '
        Me.menu_Ingreso.Name = "menu_Ingreso"
        Me.menu_Ingreso.Text = "Ingreso"
'
        'RadMenuItem4
        '
        Me.RadMenuItem4.Name = "RadMenuItem4"
Me.RadMenuItem4.Text = "Eliminacion"
        '
        'menuAprender
```

```
        '
        Me.menuAprender.Name = "menuAprender"
        Me.menuAprender.Text = "Aprender"
        '
        'informacion
'
        Me.informacion.ForeColor = System.Drawing.Color.Red
        Me.informacion.Location = New System.Drawing.Point(745, 71)
        Me.informacion.Name = "informacion"
        Me.informacion.Size = New System.Drawing.Size(430, 423)
        Me.informacion.TabIndex = 37
        Me.informacion.Text = ""
        '
        'DsAudioOut1
        '
        DsAudioOutputDevice1.DeviceName = "Altavoces (Realtek High Definit"
        DsAudioOutputDevice1.InternalData =
CType(resources.GetObject("DsAudioOutputDevice1.InternalData"), Vcl.VclBinaryData)
        Me.DsAudioOut1.AudioOutputDevice = DsAudioOutputDevice1
        SourcePin1.ConnectionData =
CType(resources.GetObject("SourcePin1.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioOut1.ClockOutputPin = SourcePin1
        StatePin1.ConnectionData = CType(resources.GetObject("StatePin1.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioOut1.EnablePin = StatePin1
        SourcePin2.ConnectionData =
CType(resources.GetObject("SourcePin2.ConnectionData"), OpenWire.PinConnections)
        DsGraph1.GraphPin = SourcePin2
        Me.DsAudioOut1.Graph = DsGraph1
```

```
        SinkPin1.ConnectionData = CType(resources.GetObject("SinkPin1.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioOut1.InputPin = SinkPin1
        Me.DsAudioOut1.InternalData =
CType(resources.GetObject("DsAudioOut1.InternalData"), Vcl.VclBinaryData)
        Me.DsAudioOut1.MasterPumping = MasterPumping1
        Me.DsAudioOut1.PrefillInterval = CType(300UI, UInteger)
        '
        'DsAudioPlayer1
        '
        SinkPin2.ConnectionData = CType(resources.GetObject("SinkPin2.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer1.ClockPin = SinkPin2
        Me.DsAudioPlayer1.Enabled = False
        StatePin2.ConnectionData = CType(resources.GetObject("StatePin2.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer1.EnablePin = StatePin2
        SinkPin3.ConnectionData = CType(resources.GetObject("SinkPin3.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer1.FileNamePin = SinkPin3
        SourcePin3.ConnectionData =
CType(resources.GetObject("SourcePin3.ConnectionData"), OpenWire.PinConnections)
        DsGraph2.GraphPin = SourcePin3
        Me.DsAudioPlayer1.Graph = DsGraph2
        Me.DsAudioPlayer1.InternalData =
CType(resources.GetObject("DsAudioPlayer1.InternalData"), Vcl.VclBinaryData)
        SourcePin4.ConnectionData =
CType(resources.GetObject("SourcePin4.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioPlayer1.OutputPin = SourcePin4
```

```
        StatePin3.ConnectionData = CType(resources.GetObject("StatePin3.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer1.PausePin = StatePin3
        StatePin4.ConnectionData = CType(resources.GetObject("StatePin4.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer1.ProgressPin = StatePin4
        Me.DsAudioPlayer1.PumpPriority = 0
        Me.DsAudioPlayer1.RestartOnNewFile = False
        SourcePin5.ConnectionData =
CType(resources.GetObject("SourcePin5.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioPlayer1.TextOutputPin = SourcePin5
        '
        'DsAudioOut2
        '
        DsAudioOutputDevice2.DeviceName = "Altavoces (Realtek High Definit"
        DsAudioOutputDevice2.InternalData =
CType(resources.GetObject("DsAudioOutputDevice2.InternalData"), Vcl.VclBinaryData)
        Me.DsAudioOut2.AudioOutputDevice = DsAudioOutputDevice2
        SourcePin6.ConnectionData =
CType(resources.GetObject("SourcePin6.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioOut2.ClockOutputPin = SourcePin6
        StatePin5.ConnectionData = CType(resources.GetObject("StatePin5.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioOut2.EnablePin = StatePin5
        SourcePin7.ConnectionData =
CType(resources.GetObject("SourcePin7.ConnectionData"), OpenWire.PinConnections)
        DsGraph3.GraphPin = SourcePin7
        Me.DsAudioOut2.Graph = DsGraph3
        SinkPin4.ConnectionData = CType(resources.GetObject("SinkPin4.ConnectionData"),
OpenWire.PinConnections)
```

```
        Me.DsAudioOut2.InputPin = SinkPin4
        Me.DsAudioOut2.InternalData =
CType(resources.GetObject("DsAudioOut2.InternalData"), Vcl.VclBinaryData)
        Me.DsAudioOut2.MasterPumping = MasterPumping2
        Me.DsAudioOut2.PrefillInterval = CType(300UI, UInteger)
        '
        'DsAudioPlayer2
        '
        SinkPin5.ConnectionData = CType(resources.GetObject("SinkPin5.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer2.ClockPin = SinkPin5
        Me.DsAudioPlayer2.Enabled = False
        StatePin6.ConnectionData = CType(resources.GetObject("StatePin6.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer2.EnablePin = StatePin6
        SinkPin6.ConnectionData = CType(resources.GetObject("SinkPin6.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer2.FileNamePin = SinkPin6
        SourcePin8.ConnectionData =
CType(resources.GetObject("SourcePin8.ConnectionData"), OpenWire.PinConnections)
        DsGraph4.GraphPin = SourcePin8
        Me.DsAudioPlayer2.Graph = DsGraph4
        Me.DsAudioPlayer2.InternalData =
CType(resources.GetObject("DsAudioPlayer2.InternalData"), Vcl.VclBinaryData)
        SourcePin9.ConnectionData =
CType(resources.GetObject("SourcePin9.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioPlayer2.OutputPin = SourcePin9
        StatePin7.ConnectionData = CType(resources.GetObject("StatePin7.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer2.PausePin = StatePin7
```

```
        StatePin8.ConnectionData = CType(resources.GetObject("StatePin8.ConnectionData"),
OpenWire.PinConnections)
        Me.DsAudioPlayer2.ProgressPin = StatePin8
        Me.DsAudioPlayer2.PumpPriority = 0
        Me.DsAudioPlayer2.RestartOnNewFile = False
        SourcePin10.ConnectionData =
CType(resources.GetObject("SourcePin10.ConnectionData"), OpenWire.PinConnections)
        Me.DsAudioPlayer2.TextOutputPin = SourcePin10
        '
        'PictureBox1
        '
        Me.PictureBox1.BackColor = System.Drawing.SystemColors.ActiveCaption
        Me.PictureBox1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
        Me.PictureBox1.Image = CType(resources.GetObject("PictureBox1.Image"),
System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(32, 30)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(397, 460)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 14
        Me.PictureBox1.TabStop = False
        '
        'principal
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(1189, 808)
Me.Controls.Add(Me.informacion)
        Me.Controls.Add(Me.menuprincipal)
```

```
        Me.Controls.Add(Me.Scope1)
        Me.Controls.Add(Me.observacion)
    Me.Controls.Add(Me.RadLabel5)
    Me.Controls.Add(Me.fuente)
    Me.Controls.Add(Me.RadLabel4)
        Me.Controls.Add(Me.RadLabel2)
        Me.Controls.Add(Me.volumen)
        Me.Controls.Add(Me.RadLabel1)
    Me.Controls.Add(Me.Tguardar)
        Me.Controls.Add(Me.Timprimir)
        Me.Controls.Add(Me.Tpdf)
        Me.Controls.Add(Me.hablado)
    Me.Controls.Add(Me.PictureBox1)
        Me.MaximizeBox = False
        Me.Name = "principal"
        '
        '
        '
        Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "principal"
        CType(Me.hablado, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Tguardar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Timprimir, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Tpdf, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.volumen, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.fuente, System.ComponentModel.ISupportInitialize).EndInit()
```

```
        CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.observacion, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Scope1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.menuprincipal, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.DsAudioOut1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.DsAudioPlayer1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.DsAudioOut2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.DsAudioPlayer2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()
    End Sub
    Friend WithEvents Temas As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents FontTIPO As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents hablado As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents Tguardar As Telerik.WinControls.UI.RadButton
    Friend WithEvents Timprimir As Telerik.WinControls.UI.RadButton
    Friend WithEvents Tpdf As Telerik.WinControls.UI.RadButton
    Friend WithEvents dletras As System.Windows.Forms.FontDialog
    Friend WithEvents PrintDialog1 As System.Windows.Forms.PrintDialog
    Friend WithEvents PrintDocument1 As System.Drawing.Printing.PrintDocument
    Friend WithEvents Save As System.Windows.Forms.SaveFileDialog
    Friend WithEvents RadMenuItem1 As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents u_Ingresar As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents u_modificar As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents ayuda As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents RadLabel1 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents volumen As Telerik.WinControls.UI.RadTrackBar
```

```
    Friend WithEvents RadLabel2 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents RadLabel4 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents fuente As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel5 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents observacion As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents Scope1 As Mitov.PlotLab.Scope
    Friend WithEvents aprender As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents color As Telerik.WinControls.RadColorDialog
    Friend WithEvents menuprincipal As Telerik.WinControls.UI.RadMenu
    Friend WithEvents menutemas As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents Menu_Letras As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents Menu_Color As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents RadMenuItem2 As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents menu_Ingreso As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents RadMenuItem4 As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents menuAprender As Telerik.WinControls.UI.RadMenuItem
    Friend WithEvents informacion As System.Windows.Forms.RichTextBox
    Friend WithEvents DsAudioOut1 As Mitov.AudioLab.DSAudioOut
    Friend WithEvents DsAudioPlayer1 As Mitov.AudioLab.DSAudioPlayer
    Friend WithEvents DsAudioOut2 As Mitov.AudioLab.DSAudioOut
    Friend WithEvents DsAudioPlayer2 As Mitov.AudioLab.DSAudioPlayer
End Class
```

Modulo

Estemódulo creo todas las Variables y objeto de tipo público que voy a usar en todo el sistema

```
Module Module1
        Objeto para manipular los usuarios
    Public usuario As New usuarios
```

```
        Objeto para manipilar la voz del sistema
    Public habla As New voz
        Objeto para manejar las reglas gramaticales
    Public regla As New reglas
        Objeto para guardar y cargar las configuraciones del sistema
    Public confi As New configuraciones
        Objecto para buscar y guardar el conocimiento del sistema
    Public infor As New conocimiento
        Objeto de los temas del sistema
    Public aqua As New Telerik.WinControls.Themes.AquaTheme
    Public vista As New Telerik.WinControls.Themes.VistaTheme
    Public breeze As New Telerik.WinControls.Themes.BreezeTheme
    Public Office2007Silver As New Telerik.WinControls.Themes.Office2007SilverTheme
    Public Windows7 As New Telerik.WinControls.Themes.Windows7Theme
    Public Office2010 As New Telerik.WinControls.Themes.Office2010Theme
    Public HighContrastBlack As New Telerik.WinControls.Themes.HighContrastBlackTheme
    Public office2007black As New Telerik.WinControls.Themes.Office2007BlackTheme
        Variable que me permite saber si recarge el sistema
    Public recarga As Integer = 0
    Public r As String
        Objeto para manipular el tipo de letra
    Public letras As Font
        Objeto que crear la pantalla principal
    Public principal As New principal
        Variable que guarda el nombre del tema actual
    Public tema_actual As String
End Module
```

Ingreso

Pantalla para el ingreso al sistema

```
Public Class ingreso
        Variable que me guarda los intentos de ingresos de los usuarios
    Dim intentos As Integer = 0
Constructor de la clase
#Region "Constructor"
    Public Sub New()
        InitializeComponent()
    End Sub
#End Region

Procedimiento principal donde invoco a la creación de las reglas gramaticales y configuración de sistema
Private Sub f_ingreso_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
Try
                Invoco al método cargar de la clase reglas para crear las reglas gramaticales del sistema
        regla.cargar()
                Invoco al método cargar de la clase configuraciones para cargar las configuraciones del sistema
        confi.cargar()
                Obtengo el tipo de letra del sistema
        Dim letra As New Font(confi.tipo_letra, confi.tamaño_letra)
        letras = letra
        Me.ThemeName = confi.tema
Me.tusuario.Focus()
        Me.RadPanel1.ThemeName = confi.tema
```

```
Me.Aceptar.ThemeName = confi.tema
        Me.cerrar.ThemeName = confi.tema
        Me.tcontra.ThemeName = confi.tema
        Me.tusuario.ThemeName = confi.tema
Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
  End Sub
```

Procedimiento que me permite validar si el usuario existe en el sistema y cuando intentos lleva el usuario por ingresar

```
Private Sub Aceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Aceptar.Click
    Try
        If tusuario.Text <> "" And tcontra.Text <> "" Then
            If usuario.buscar(Me.tusuario.Text, Me.tcontra.Text) = 1 Then
                principal.Show()
                Me.Hide()
            Else
                intentos += 1
                If intentos = 3 Then
MsgBox("Intentos Fallidos")
                    End
                Else
                    habla.leer("Usuario No Valido")
tusuario.Text = ""
                    tcontra.Text = ""
                End If
            End If
        Else
```

```
            habla.leer("Faltan Datos")
        End If
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
```

Procedimiento que me permite cerrar la pantalla de ingreso

```
Private Sub cerrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cerrar.Click
    End
End Sub
End Class
```

Diseño

Estasección se genera automáticamentecuando se esta diseñando la pantalla

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class ingreso
    Inherits Telerik.WinControls.UI.RadForm
    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
```

```
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.Aceptar = New Telerik.WinControls.UI.RadButton()
        Me.cerrar = New Telerik.WinControls.UI.RadButton()
        Me.RadLabel1 = New Telerik.WinControls.UI.RadLabel()
        Me.RadLabel2 = New Telerik.WinControls.UI.RadLabel()
        Me.tusuario = New Telerik.WinControls.UI.RadTextBox()
        Me.tcontra = New Telerik.WinControls.UI.RadTextBox()
        Me.DesertTheme1 = New Telerik.WinControls.Themes.DesertTheme()
        Me.RadPanel1 = New Telerik.WinControls.UI.RadPanel()
        Me.PictureBox1 = New System.Windows.Forms.PictureBox()
        CType(Me.Aceptar, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.cerrar, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.tusuario, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.tcontra, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadPanel1, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.RadPanel1.SuspendLayout()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
```

```
        '
        'Aceptar
        '
        Me.Aceptar.Location = New System.Drawing.Point(46, 184)
        Me.Aceptar.Name = "Aceptar"
        Me.Aceptar.Size = New System.Drawing.Size(119, 31)
Me.Aceptar.TabIndex = 0
        Me.Aceptar.Text = "Aceptar"
        Me.Aceptar.ThemeName = "Desert"
        '
        'cerrar
'
        Me.cerrar.Location = New System.Drawing.Point(255, 184)
        Me.cerrar.Name = "cerrar"
        Me.cerrar.Size = New System.Drawing.Size(119, 31)
Me.cerrar.TabIndex = 1
        Me.cerrar.Text = "Cerrar"
Me.cerrar.ThemeName = "Desert"
        '
        'RadLabel1
        '
        Me.RadLabel1.Location = New System.Drawing.Point(182, 10)
        Me.RadLabel1.Name = "RadLabel1"
        Me.RadLabel1.Size = New System.Drawing.Size(44, 18)
        Me.RadLabel1.TabIndex = 2
        Me.RadLabel1.Text = "Usuario"
        '
        'RadLabel2
        '
        Me.RadLabel2.Location = New System.Drawing.Point(182, 66)
```

```
        Me.RadLabel2.Name = "RadLabel2"
        Me.RadLabel2.Size = New System.Drawing.Size(63, 18)
Me.RadLabel2.TabIndex = 3
        Me.RadLabel2.Text = "Contraseña"
        '
        'tusuario
'
        Me.tusuario.Location = New System.Drawing.Point(255, 9)
        Me.tusuario.Name = "tusuario"
        Me.tusuario.Size = New System.Drawing.Size(218, 20)
        Me.tusuario.TabIndex = 4
        Me.tusuario.TabStop = False
        Me.tusuario.ThemeName = "Desert"
        '
        'tcontra
        '
        Me.tcontra.Location = New System.Drawing.Point(255, 64)
        Me.tcontra.Name = "tcontra"
        Me.tcontra.PasswordChar = Global.Microsoft.VisualBasic.ChrW(42)
        Me.tcontra.Size = New System.Drawing.Size(212, 20)
        Me.tcontra.TabIndex = 5
        Me.tcontra.TabStop = False
        Me.tcontra.ThemeName = "Desert"
        '
        'RadPanel1
        '
        Me.RadPanel1.Controls.Add(Me.PictureBox1)
        Me.RadPanel1.Controls.Add(Me.RadLabel1)
        Me.RadPanel1.Controls.Add(Me.tcontra)
        Me.RadPanel1.Controls.Add(Me.Aceptar)
```

```
        Me.RadPanel1.Controls.Add(Me.tusuario)
        Me.RadPanel1.Controls.Add(Me.cerrar)
        Me.RadPanel1.Controls.Add(Me.RadLabel2)
        Me.RadPanel1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.RadPanel1.Location = New System.Drawing.Point(0, 0)
        Me.RadPanel1.Name = "RadPanel1"
        Me.RadPanel1.Size = New System.Drawing.Size(479, 232)
        Me.RadPanel1.TabIndex = 6
        '
        'PictureBox1
        '
        Me.PictureBox1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
        Me.PictureBox1.Image = Global.SEUZ.My.Resources.Resources.contraseña100
        Me.PictureBox1.Location = New System.Drawing.Point(24, 10)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(141, 143)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 6
        Me.PictureBox1.TabStop = False
        '
        'ingreso
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(479, 232)
        Me.Controls.Add(Me.RadPanel1)
        Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedToolWindow
Me.Name = "ingreso"
        '
```

```
        '
        '
Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "Ingreso"
        Me.ThemeName = "Desert"
        CType(Me.Aceptar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.cerrar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.tusuario, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.tcontra, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadPanel1, System.ComponentModel.ISupportInitialize).EndInit()
        Me.RadPanel1.ResumeLayout(False)
        Me.RadPanel1.PerformLayout()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
    End Sub
    Friend WithEvents Aceptar As Telerik.WinControls.UI.RadButton
    Friend WithEvents cerrar As Telerik.WinControls.UI.RadButton
    Friend WithEvents RadLabel1 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents RadLabel2 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents tusuario As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents tcontra As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents DesertTheme1 As Telerik.WinControls.Themes.DesertTheme
    Friend WithEvents RadPanel1 As Telerik.WinControls.UI.RadPanel
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
End Class
```

Ingresar usuario

Pantalla para ingreso de Nuevo Usuario.

```
Librerías que me permiten interactuar con el reconocimiento de voz y la sintetización de voz
Imports System.Speech.Recognition
Imports System.Speech.Recognition.SrgsGrammar
Public Class ingresar_usuario
        Propiedad que me permite manejar el reconocimiento de voz
    Private WithEvents engSR As New System.Speech.Recognition.SpeechRecognizer
        Propiedad que me permite manejar la sintetizacion de voz
    Private WithEvents engSS As New System.Speech.Synthesis.SpeechSynthesizer
Procedimiento que me permite validar lo que ha ingresado el usuario si falta algun dato el
sistema se lo dirá en modo voz
    Public Sub ingresar()
        If (nick.Text = "") Then
habla.leer("Ingrese usuario")
            nick.Focus()
Exit Sub
        End If
        If (key.Text = "") Then
habla.leer("Ingrese Clave")
            key.Focus()
Exit Sub
        End If
        If (key.Text.Length > 8) Then
habla.leer("Ingrese una Clave con más de 8 caracteres")
rkey.Focus()
        End If
        If rkey.Text = "" Then
habla.leer("Ingrese nuevamenmte la clave")
```

```
rkey.Focus()
            Exit Sub
        End If
        If (key.Text <> rkey.Text) Then
habla.leer("clave no coinciden")
Exit Sub
        End If
        If (nombre.Text = "") Then
habla.leer("ingre el nombre del usuario")
nombre.Focus()
            Exit Sub
        End If
If (apellido.Text = "") Then
            habla.leer("ingrese el apellido del usuario")
apellido.Focus()
            Exit Sub
        End If
Si no faltan datos proceso a guardar los datos en la propiedades del objeto usuario
        usuario.nick = nick.Text
        usuario.clave = key.Text
usuario.Nombre = nombre.Text
        usuario.Apellido = apellido.Text
        If (usuario.grabar = 0) Then
            habla.leer("Usuario Nuevo Ingresado ")
nick.Text = ""
            key.Text = ""
            rkey.Text = ""
nombre.Text = ""
            apellido.Text = ""
        Else
```

```
            habla.leer("usuario ya existe")
End If
   End Sub
Procedimiento que me permite llamar al procedimiento ingresar
   Private Sub grabar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles grabar.Click
      ingresar()
   End Sub

Procedimiento que me permite librerar los recursos utilizados por esta ventana
Private Sub ingresar_usuario_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
      Dispose()
      engSR.Dispose()
   End Sub

Procedimiento que me permite cargar las configuraciones del sistema para luego ser aplicadas
a la ventana actual
Private Sub ingresar_usuario_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
      Me.ThemeName = confi.tema
      Me.nick.ThemeName = confi.tema
      Me.RadPanel1.ThemeName = confi.tema
      Me.apellido.ThemeName = confi.tema
Me.nombre.ThemeName = confi.tema
      Me.rkey.ThemeName = confi.tema
      Me.key.ThemeName = confi.tema
      Me.grabar.ThemeName = confi.tema
      Me.cancelar.ThemeName = confi.tema
      ' creo los comandos con los que voy a trabajar
```

```
Dim itemA As New Speech.Recognition.SrgsGrammar.SrgsItem("nick")
        Dim itemB As New Speech.Recognition.SrgsGrammar.SrgsItem("clave")
        Dim itemC As New Speech.Recognition.SrgsGrammar.SrgsItem("repertir clave")
        Dim itemD As New Speech.Recognition.SrgsGrammar.SrgsItem("nombre")
        Dim itemE As New Speech.Recognition.SrgsGrammar.SrgsItem("apellido")
        Dim itemF As New Speech.Recognition.SrgsGrammar.SrgsItem("grabar")
‘creo la regla con los comandos que tengo q incorporar
Dim rule As New Speech.Recognition.SrgsGrammar.SrgsRule("Comandos", _
          New Speech.Recognition.SrgsGrammar.SrgsOneOf(itemA, itemB, itemC, itemD,
itemE, itemF))
‘creo el documento de ejemplo
Dim doc As New Speech.Recognition.SrgsGrammar.SrgsDocument
‘agrego la regla y la pongo como regla por defecto
doc.Rules.Add(rule)
        doc.Root = rule
‘cargo la gramatica de la aplicacion
engSR.LoadGrammar(New Speech.Recognition.Grammar(doc))
    End Sub

Procedimiento que me permite reconocer el comando que le dicte al sistema
    Private Sub engSR_SpeechRecognized(ByVal sender As Object, ByVal e As
System.Speech.Recognition.SpeechRecognizedEventArgs) Handles
engSR.SpeechRecognized
        Select Case e.Result.Text
            Case "grabar"
ingresar()
            Case ("nick")
                Me.nick.Focus()
            Case "repetir clave"
                Me.rkey.Focus()
```

```
            Case "clave"
                Me.key.Focus()
            Case "nombre"
                Me.nombre.Focus()
            Case "apellido"
                Me.apellido.Focus()
End Select
    End Sub

Procemiento que me permite cerrar la ventana actual
Private Sub cancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cancelar.Click
        Me.Close()
    End Sub
End Class

Diseño
Esta parte se genera automáticamente cuando se esta diseñando la ventana

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class ingresar_usuario
    Inherits Telerik.WinControls.UI.RadForm

    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
```

```
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.nick = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel1 = New Telerik.WinControls.UI.RadLabel()
        Me.nombre = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel2 = New Telerik.WinControls.UI.RadLabel()
        Me.RadLabel3 = New Telerik.WinControls.UI.RadLabel()
        Me.RadLabel4 = New Telerik.WinControls.UI.RadLabel()
        Me.key = New Telerik.WinControls.UI.RadTextBox()
        Me.rkey = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel5 = New Telerik.WinControls.UI.RadLabel()
        Me.apellido = New Telerik.WinControls.UI.RadTextBox()
        Me.grabar = New Telerik.WinControls.UI.RadButton()
        Me.cancelar = New Telerik.WinControls.UI.RadButton()
        Me.RadPanel1 = New Telerik.WinControls.UI.RadPanel()
        Me.PictureBox1 = New System.Windows.Forms.PictureBox()
        CType(Me.nick, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.nombre, System.ComponentModel.ISupportInitialize).BeginInit()
```

```
CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.RadLabel3, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.key, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.rkey, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.apellido, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.grabar, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.cancelar, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.RadPanel1, System.ComponentModel.ISupportInitialize).BeginInit()
Me.RadPanel1.SuspendLayout()
CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
'
'nick
'
Me.nick.Location = New System.Drawing.Point(302, 12)
Me.nick.Name = "nick"
Me.nick.Size = New System.Drawing.Size(231, 20)
Me.nick.TabIndex = 0
Me.nick.TabStop = False
'
'RadLabel1
'
Me.RadLabel1.Location = New System.Drawing.Point(217, 12)
Me.RadLabel1.Name = "RadLabel1"
Me.RadLabel1.Size = New System.Drawing.Size(31, 18)
Me.RadLabel1.TabIndex = 1
Me.RadLabel1.Text = "NICK"
```

```
'
'nombre
'
Me.nombre.Location = New System.Drawing.Point(302, 139)
Me.nombre.Name = "nombre"
Me.nombre.Size = New System.Drawing.Size(264, 20)
Me.nombre.TabIndex = 2
Me.nombre.TabStop = False
'
'RadLabel2
'
Me.RadLabel2.Location = New System.Drawing.Point(217, 141)
Me.RadLabel2.Name = "RadLabel2"
Me.RadLabel2.Size = New System.Drawing.Size(52, 18)
Me.RadLabel2.TabIndex = 3
Me.RadLabel2.Text = "NOMBRE"
'
'RadLabel3
'
Me.RadLabel3.Location = New System.Drawing.Point(217, 55)
Me.RadLabel3.Name = "RadLabel3"
Me.RadLabel3.Size = New System.Drawing.Size(39, 18)
Me.RadLabel3.TabIndex = 4
Me.RadLabel3.Text = "CLAVE"
'
'RadLabel4
'
Me.RadLabel4.Location = New System.Drawing.Point(217, 96)
Me.RadLabel4.Name = "RadLabel4"
Me.RadLabel4.Size = New System.Drawing.Size(82, 18)
```

```
Me.RadLabel4.TabIndex = 5
        Me.RadLabel4.Text = "REPETIR CLAVE"
'
        'key
        '
        Me.key.Location = New System.Drawing.Point(302, 53)
        Me.key.Name = "key"
        Me.key.PasswordChar = Global.Microsoft.VisualBasic.ChrW(42)
        Me.key.Size = New System.Drawing.Size(139, 20)
        Me.key.TabIndex = 6
        Me.key.TabStop = False
        '
        'rkey
        '
        Me.rkey.Location = New System.Drawing.Point(302, 94)
        Me.rkey.Name = "rkey"
        Me.rkey.PasswordChar = Global.Microsoft.VisualBasic.ChrW(42)
        Me.rkey.Size = New System.Drawing.Size(155, 20)
        Me.rkey.TabIndex = 7
        Me.rkey.TabStop = False
        '
        'RadLabel5
        '
        Me.RadLabel5.Location = New System.Drawing.Point(217, 184)
        Me.RadLabel5.Name = "RadLabel5"
        Me.RadLabel5.Size = New System.Drawing.Size(56, 18)
Me.RadLabel5.TabIndex = 8
        Me.RadLabel5.Text = "APELLIDO"
'
        'apellido
```

```
        '
        Me.apellido.Location = New System.Drawing.Point(302, 182)
        Me.apellido.Name = "apellido"
        Me.apellido.Size = New System.Drawing.Size(264, 20)
Me.apellido.TabIndex = 9
        Me.apellido.TabStop = False
'
        'grabar
        '
        Me.grabar.Location = New System.Drawing.Point(12, 270)
        Me.grabar.Name = "grabar"
        Me.grabar.Size = New System.Drawing.Size(107, 36)
Me.grabar.TabIndex = 10
        Me.grabar.Text = "Grabar"
'
        'cancelar
        '
        Me.cancelar.Location = New System.Drawing.Point(302, 270)
        Me.cancelar.Name = "cancelar"
        Me.cancelar.Size = New System.Drawing.Size(107, 36)
        Me.cancelar.TabIndex = 11
        Me.cancelar.Text = "Cancelar"
        '
        'RadPanel1
        '
        Me.RadPanel1.Controls.Add(Me.PictureBox1)
        Me.RadPanel1.Controls.Add(Me.cancelar)
        Me.RadPanel1.Controls.Add(Me.nick)
        Me.RadPanel1.Controls.Add(Me.grabar)
        Me.RadPanel1.Controls.Add(Me.RadLabel1)
```

```
        Me.RadPanel1.Controls.Add(Me.apellido)
        Me.RadPanel1.Controls.Add(Me.nombre)
        Me.RadPanel1.Controls.Add(Me.RadLabel5)
        Me.RadPanel1.Controls.Add(Me.RadLabel2)
        Me.RadPanel1.Controls.Add(Me.rkey)
        Me.RadPanel1.Controls.Add(Me.RadLabel3)
        Me.RadPanel1.Controls.Add(Me.key)
        Me.RadPanel1.Controls.Add(Me.RadLabel4)
        Me.RadPanel1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.RadPanel1.Location = New System.Drawing.Point(0, 0)
        Me.RadPanel1.Name = "RadPanel1"
        Me.RadPanel1.Size = New System.Drawing.Size(576, 331)
        Me.RadPanel1.TabIndex = 12
        '
        'PictureBox1
        '
        Me.PictureBox1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
        Me.PictureBox1.Image = Global.SEUZ.My.Resources.Resources.Nuevo_usuario
        Me.PictureBox1.Location = New System.Drawing.Point(12, 12)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(163, 171)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 12
        Me.PictureBox1.TabStop = False
        '
        'ingresar_usuario
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
```

```
        Me.ClientSize = New System.Drawing.Size(576, 331)
        Me.Controls.Add(Me.RadPanel1)
Me.Name = "ingresar_usuario"
        '
        '
'
        Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "ingresar_usuario"
        Me.TopMost = True
        CType(Me.nick, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.nombre, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel3, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.key, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.rkey, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.apellido, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.grabar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.cancelar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadPanel1, System.ComponentModel.ISupportInitialize).EndInit()
        Me.RadPanel1.ResumeLayout(False)
        Me.RadPanel1.PerformLayout()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)

    End Sub
```

```
    Friend WithEvents nick As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel1 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents nombre As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel2 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents RadLabel3 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents RadLabel4 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents key As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents rkey As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel5 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents apellido As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents grabar As Telerik.WinControls.UI.RadButton
    Friend WithEvents cancelar As Telerik.WinControls.UI.RadButton
    Friend WithEvents RadPanel1 As Telerik.WinControls.UI.RadPanel
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
End Class
```

Clase conocimiento

Clase que me permite buscar el conocimiento del sistema que está guardado en la base de datos

```
Librería que me permite manipular la base de datos
Imports MySql.Data.MySqlClient

Public Class conocimiento
Atributos del tema a buscar en este caso el conocimiento a buscar
    Private ctemas As String
    Private ccategoria As String
    Private cfecha As Date
    Private ccontenido As String
    Private cfuente As String
```

```
    Private cobservacion As String
    Private conexion As New MySqlConnection
    'creamos una variable para las consultas
    Private consultas As String
    Private comandos As New MySqlCommand
```

Método que me permite buscar el conocimiento guardado en la base de datos

```
    Public Function buscar(ByVal cadena As String) As Integer
        Try
            Dim temas As String = ""
            conexion.ConnectionString =
"server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
            conexion.Open()
            consultas = "select tema from seuz.conocimiento"
            comandos.Connection = conexion
            comandos.CommandText = consultas
            'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
            comandos.CommandType = CommandType.Text
            'Creamos un lector de datos (IDataReader) y lo inicializamos
            'con el lector del objeto MySQLCommand
Dim dr As System.Data.IDataReader
dr = comandos.ExecuteReader()
            'combinamos las palabras con el conocimiento
Dim conocimiento As New ArrayList
            While dr.Read
conocimiento.Add(dr("tema"))
            End While
dr.Dispose()
            For i = 0 To conocimiento.Count - 1
```

```
            If (cadena.Contains(conocimiento.Item(i).ToString.ToLower)) Then
               temas = conocimiento.Item(i).ToString
               Exit For
            End If
         Next
         conexion.Close()
         conexion.Open()
'creamos una consulta para la base de datos
         consultas = "select * from seuz.conocimiento where tema='" + temas + "' "
         'establecemos la conexion
         'comandos.Connection = conexion
         comandos.CommandText = consultas
         'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
         comandos.CommandType = CommandType.Text
         'Creamos un lector de datos (IDataReader) y lo inicializamos
         'con el lector del objeto MySQLCommand
Dim datos As System.Data.IDataReader
datos = comandos.ExecuteReader()
         'Mientras haya datos para leer los mostramos
         While datos.Read()
            'Al igual de los objetos DataRow, la clase IDataReader también tiene
            'un método por defecto .Item que funciona exactamente igual
            ctemas = datos("tema")
            ccontenido = datos("descripcion")
            cfuente = datos("fuente")
            cobservacion = datos("observacion")
         End While
         'Cerramos la conexión con el servidor
conexion.Close()
      Catch ex As Exception
```

```
            MsgBox(ex.ToString)
        End Try


    End Function
```

Procedimiento que me permite guardar un nuevo conocimiento a la base de datos

```
    Public Sub guardar(ByVal id As Integer, ByVal categoria As Integer)
        Try
            conexion.ConnectionString =
"server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
conexion.Open()
            comandos.Connection = conexion
            comandos.CommandText = "SELECT max(id)+1 as maximo FROM
seuz.conocimiento"
            comandos.CommandType = CommandType.Text
            Dim dr As System.Data.IDataReader
            dr = comandos.ExecuteReader()
            Dim ultimo As Integer
            While dr.Read
                ultimo = dr("maximo")
            End While
            dr.Dispose()
            Dim fecha As String
            fecha = Now().Year.ToString + "-" + Now.Month.ToString + "-" + Now.Day.ToString
+ " " + Now.Hour.ToString + ":" + Now.Minute.ToString + ":" + Now.Second.ToString
comandos.CommandText = "insert into
conocimiento(id,tema,descripcion,id_usuario,fecha_cre,id_categoria,fecha_mo,fuente,observa
cion,estado)values(" + ultimo.ToString + ",'" + ctemas + "','" + ccontenido + "'," &
```

```
id.ToString & "'" & fecha & "'," & categoria & ",'" + fecha + "','" + cfuente + "','" +
cobservacion + "','A')"
comandos.CommandType = CommandType.Text
            comandos.ExecuteNonQuery()
conexion.Close()
            habla.leer("Informacion Grabada")
Catch ex As Exception
            conexion.Close()
            MsgBox(ex.ToString)
        End Try
         End Sub

Metodo que me permite obtener u establer el nombre del tema del conocimiento
    Public Property tema() As String
        ' bloque Get para devolver
‘el valor de la propiedad
Get
            Return ctemas
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            ctemas = Value
        End Set
    End Property

Metodo que me permite obtener u establecer la categoria del conocimiento
    Public Property categoria() As Integer
' bloque Get para devolver
        ' el valor de la propiedad
```

```
Get
        Return ccategoria
    End Get
    ' bloque Set para asignar
' valor a la propiedad
    Set(ByVal Value As Integer)
ccategoria = Value
    End Set
  End Property

Metodo que me permite obtener u establecer la fecha que va a ser ingresado el conocimiento
Public Property fecha() As Date
    ' bloque Get para devolver
    ' el valor de la propiedad
Get
        Return cfecha
    End Get
    ' bloque Set para asignar
    ' valor a la propiedad
    Set(ByVal Value As Date)
      cfecha = Value
    End Set
  End Property

Metodo que me permite obtener u establecer el contenido del conocimiento
  Public Property contenido() As String
' bloque Get para devolver
    ' el valor de la propiedad
Get
        Return ccontenido
```

```
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            ccontenido = Value
        End Set
    End Property

Metodo que me permite obtener u establecer la fuente del conocimiento
    Public Property fuente() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return cfuente
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            cfuente = Value
        End Set
    End Property

Metodo que me permite obtener u establecer la observaciones del conocimiento
    Public Property observacion() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return cobservacion
        End Get
        ' bloque Set para asignar
```

```
        ' valor a la propiedad
        Set(ByVal Value As String)
            cobservacion = Value
        End Set
    End Property
End Class
```

Clase configuraciones

Clase que me permite cargar las configuraciones que tengo del sistema como el tema y tipo de letra

```
Librería que me permite manipular la base de datos
Imports MySql.Data.MySqlClient
Public Class configuraciones
        Atributos de la configuraciones del sistema
    Private ctema As String
    Private ctipo_Letra As String
    Private ctamaño_letra As Integer
    Private ccolor As String
'La variable bConexionExitosa nos permitirá finalizar el
    'programa si la conexión falla
    Private bConexionExitosa As Boolean = True
    'Creamos un objeto de tipo Connection y configuramos los parámetros de la conexión
Private conexion As New MySqlConnection
    'creamos una variable para las consultas
    Private consultas As String
    Private comandos As New MySqlCommand

Metodo para cargar las configuraciones del sistema
    Public Sub cargar()
```

```
        Try
            'Los parámetros de la sobrecarga con más parámetros son:
            '1. Dirección IP o nombre de la máquina con el servidor de MySQLS
            '2. Nombre de la base de datos
            '3. Nombre de usuario con acceso a la base de datos señalada anteriormente
            '4. Contraseña para el nombre de usuario citado
conexion.ConnectionString = "server=localhost ;" & "database=seuz;" & "user id=root;" &
"password=dar"
'Abrimos la conexión y comprobamos que no hay error
If conexion.State = ConnectionState.Open Then

Else
                conexion.Open()
            End If
            'creamos una consulta para la base de datos
            consultas = "select * from seuz.configuracion"
            'establecemos la conexion
            comandos.Connection = conexion
            comandos.CommandText = consultas
            'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
            comandos.CommandType = CommandType.Text
            'Creamos un lector de datos (IDataReader) y lo inicializamos
            'con el lector del objeto MySQLCommand
Dim dr As System.Data.IDataReader
dr = comandos.ExecuteReader()
            'Mientras haya datos para leer los mostramos
            While dr.Read()
                'Al igual de los objetos DataRow, la clase IDataReader también tiene
                'un método por defecto .Item que funciona exactamente igual
                ctema = dr("tema")
```

```
            ctamaño_letra = dr("tama_Letra")
            ctipo_Letra = dr("tipo_letra")
            ccolor = dr("color")
        End While
        'Cerramos la conexión con el servidor
conexion.Close()
    Catch ex As Exception
'mostramos los errores si susedieran
        MsgBox(ex.ToString, MsgBoxStyle.Information)
    End Try
  End Sub

Metodo para cambiar las configuraciones del sistema
Public Sub establecer_letra(ByVal tamaño As Integer, ByVal letra As String, ByVal color As
String)
    Try
    conexion.ConnectionString = "server=localhost ;" & "database=seuz;" & "user id=root;"
& "password=dar"
    conexion.Open()
    comandos.Connection = conexion
    comandos.CommandText = "update configuracion set  tama_letra=" + tamaño.ToString
+ ",tipo_Letra='" + letra + "',color='" + color + "' where id=1"
    comandos.CommandType = CommandType.Text
    comandos.ExecuteNonQuery()
    conexion.Close()
    Catch ex As Exception
MsgBox(ex.ToString, MsgBoxStyle.Information)
End Try
End Sub
```

Metodo para cambiar el campo errores que indica si sale la ventana de errores del recocimiento de voz

```
Public Sub cambiar_error()
        conexion.ConnectionString = "server=localhost ;" & "database=seuz;" & "user id=root;" & "password=dar"
        conexion.Open()
        comandos.Connection = conexion
        comandos.CommandText = "update configuracion set  errores=" + cerror.ToString + " where id=1"
        comandos.CommandType = CommandType.Text
        comandos.ExecuteNonQuery()
        conexion.Close()
    End Sub
```

Metodo que me permite guardar el nombre del tema que escogi en la pantalla temas

```
    Public Sub temas(ByVal temas As String)
        Try
            conexion.ConnectionString = "server=localhost ;" & "database=seuz;" & "user id=root;" & "password=dar"
            conexion.Open()
            comandos.Connection = conexion
            comandos.CommandText = "update configuracion set  tema='" + temas + "' where id=1"
            comandos.CommandType = CommandType.Text
            comandos.ExecuteNonQuery()
            conexion.Close()
        Catch ex As Exception
            MsgBox(ex.ToString, MsgBoxStyle.Information)
        End Try
    End Sub
```

Metodo que me permite obtener u establecer el nombre del tema del sistema

```
Public Property tema() As String
```

```
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return ctema
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            ctema = Value
        End Set
    End Property

Metodo que me permite obtener u establecer el color de la letra del sistema
    Public Property color() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return ccolor
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            ccolor = Value
        End Set
    End Property

Metodo que me devuelve un uno o cero que indica si se muestran los errores
Public Property errores() As String
        ' bloque Get para devolver
        ' el valor de la propiedad
```

```
        Get
            Return cerror
        End Get
        ' bloque Set para asignar
        ' valor a la propiedad
        Set(ByVal Value As String)
            cerror = Value
        End Set
    End Property

Metodo que me permite obtener u establecer el tipo de letra del sistema
    Public Property tipo_letra() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return ctipo_Letra
        End Get
' bloque Set para asignar
        ' valor a la propiedad
Set(ByVal Value As String)
            ctipo_Letra = Value
        End Set
    End Property

Metodo que me permite obtener u establecer el tamaño de la letra del sistema
    Public Property tamaño_letra() As String
' bloque Get para devolver
        ' el valor de la propiedad
Get
            Return ctamaño_letra
```

```
        End Get
' bloque Set para asignar
        ' valor a la propiedad
Set(ByVal Value As String)
            ctamaño_letra = Value
        End Set
    End Property
End Class
```

Pantalla bienvenida

Librería que me permite manipular directorios en el sistema operativo

```
Imports System.IO

Esta es la primer pantalla que aparece
Public Class bienvenida

    Private Sub bienvenida_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

        tiempo.Start()
                En esta parte verifico si no existe los directorios que utiliza el sistema
        If Not Directory.Exists("c:\SEUZ") Then
            Directory.CreateDirectory("c:\SEUZ")
        End If
        If Not Directory.Exists("C:\SEUZ\DOCUMENTOS") Then
            Directory.CreateDirectory("c:\SEUZ\DOCUMENTOS")
        End If
    End Sub
Procedimiento para manipular el progress bar de esta pantalla
```

```
Private Sub tiempo_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles
tiempo.Tick
        If progreso.Value1 < 100 Then
            progreso.Value1 += 25
            progreso.Refresh()
        ElseIf progreso.Value1 = 100 Then
            tiempo.Stop()
            Dim ingreso As New ingreso
            ingreso.Show()
            Me.Hide()
        End If
    End Sub
End Class
```

Diseño

Esta parte se genera automaticamente cuando se diseña la pantalla de bienvenida

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class bienvenida
    Inherits Telerik.WinControls.UI.RadForm

    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
```

```
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(bienvenida))
        Me.PictureBox1 = New System.Windows.Forms.PictureBox()
        Me.progreso = New Telerik.WinControls.UI.RadProgressBar()
        Me.tiempo = New System.Windows.Forms.Timer(Me.components)
        Me.DesertTheme1 = New Telerik.WinControls.Themes.DesertTheme()
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.progreso, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'PictureBox1
        '
        Me.PictureBox1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.PictureBox1.Image = CType(resources.GetObject("PictureBox1.Image"),
System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(3, 3)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(876, 785)
```

```
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 0
        Me.PictureBox1.TabStop = False
        '
        'progreso
        '
        Me.progreso.Dash = False
        Me.progreso.Location = New System.Drawing.Point(254, 400)
        Me.progreso.Name = "progreso"
        Me.progreso.SeparatorColor1 = System.Drawing.Color.FromArgb(CType(CType(25,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(71, Byte), Integer),
CType(CType(216, Byte), Integer))
        Me.progreso.SeparatorColor2 = System.Drawing.Color.FromArgb(CType(CType(20,
Byte), Integer), CType(CType(0, Byte), Integer), CType(CType(71, Byte), Integer),
CType(CType(216, Byte), Integer))
        Me.progreso.SeparatorWidth = 7
        Me.progreso.ShowProgressIndicators = True
        Me.progreso.Size = New System.Drawing.Size(362, 27)
        Me.progreso.StepWidth = 14
        Me.progreso.TabIndex = 1
        Me.progreso.Text = "0%"
        Me.progreso.TextAlignment = System.Drawing.ContentAlignment.MiddleCenter
Me.progreso.ThemeName = "Desert"
        '
        'tiempo
        '
        Me.tiempo.Interval = 1000
        '
        'bienvenida
```

```
'
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(882, 791)
        Me.Controls.Add(Me.progreso)
        Me.Controls.Add(Me.PictureBox1)
        Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
        Me.Name = "bienvenida"
        Me.Padding = New System.Windows.Forms.Padding(3)
        '
        '
        '
        Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "bienvenida"
        Me.ThemeName = "Desert"
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.progreso, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)

    End Sub
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents progreso As Telerik.WinControls.UI.RadProgressBar
    Friend WithEvents tiempo As System.Windows.Forms.Timer
    Friend WithEvents DesertTheme1 As Telerik.WinControls.Themes.DesertTheme
End Class
```

Pantalla de Ayuda

Pantalla de información del sistema y describe el sistema y los derechos de autor

```
Imports Microsoft.VisualBasic
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Drawing
Imports System.Windows.Forms
Imports System.Reflection

Public Partial Class ayuda
    Inherits Telerik.WinControls.UI.RadForm

        Public Sub New()
            InitializeComponent()

            ' Initialize the AboutBox to display the product information from the assembly
information.
            ' Change assembly information settings for your application through either:
            ' - Project->Properties->Application->Assembly Information
            ' - AssemblyInfo.cs
            Me.Text = String.Format("About {0}", AssemblyTitle)
            Me.radLabelProductName.Text = AssemblyProduct
            Me.radLabelVersion.Text = String.Format("Version {0}", AssemblyVersion)
            Me.radLabelCopyright.Text = AssemblyCopyright
            Me.radLabelCompanyName.Text = AssemblyCompany
            Me.radTextBoxDescription.Text = AssemblyDescription
        End Sub
```

```
        #Region "Assembly Attribute Accessors"

        Public ReadOnly Property AssemblyTitle() As String
            Get
                ' Get all Title attributes on this assembly
                Dim attributes As Object() =
System.Reflection.Assembly.GetExecutingAssembly().GetCustomAttributes(GetType(Assem
blyTitleAttribute), False)
                ' If there is at least one Title attribute
                If attributes.Length > 0 Then
                    ' Select the first one
                    Dim titleAttribute As AssemblyTitleAttribute =
CType(attributes(0), AssemblyTitleAttribute)
                    ' If it is not an empty string, return it
                    If titleAttribute.Title <> "" Then
                        Return titleAttribute.Title
                    End If
                End If
                ' If there was no Title attribute, or if the Title attribute was the empty
string, return the .exe name
                Return
System.IO.Path.GetFileNameWithoutExtension(System.Reflection.Assembly.GetExecutingA
ssembly().CodeBase)
            End Get
        End Property

        Public ReadOnly Property AssemblyVersion() As String
            Get
```

```
                Return
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString()
            End Get
        End Property

        Public ReadOnly Property AssemblyDescription() As String
            Get
                ' Get all Description attributes on this assembly
                Dim attributes As Object() =
System.Reflection.Assembly.GetExecutingAssembly().GetCustomAttributes(GetType(Assem
blyDescriptionAttribute), False)
                ' If there aren't any Description attributes, return an empty string
                If attributes.Length = 0 Then
                    Return ""
                End If
                ' If there is a Description attribute, return its value
                Return (CType(attributes(0),
AssemblyDescriptionAttribute)).Description
            End Get
        End Property

        Public ReadOnly Property AssemblyProduct() As String
            Get
                ' Get all Product attributes on this assembly
                Dim attributes As Object() =
System.Reflection.Assembly.GetExecutingAssembly().GetCustomAttributes(GetType(Assem
blyProductAttribute), False)
                ' If there aren't any Product attributes, return an empty string
                If attributes.Length = 0 Then
                    Return ""
```

```
                End If
                ' If there is a Product attribute, return its value
                Return (CType(attributes(0), AssemblyProductAttribute)).Product
            End Get
        End Property

        Public ReadOnly Property AssemblyCopyright() As String
            Get
                ' Get all Copyright attributes on this assembly
                Dim attributes As Object() =
System.Reflection.Assembly.GetExecutingAssembly().GetCustomAttributes(GetType(Assem
blyCopyrightAttribute), False)
                ' If there aren't any Copyright attributes, return an empty string
                If attributes.Length = 0 Then
                    Return ""
                End If
                ' If there is a Copyright attribute, return its value
                Return (CType(attributes(0), AssemblyCopyrightAttribute)).Copyright
            End Get
        End Property

        Public ReadOnly Property AssemblyCompany() As String
            Get
                ' Get all Company attributes on this assembly
                Dim attributes As Object() =
System.Reflection.Assembly.GetExecutingAssembly().GetCustomAttributes(GetType(Assem
blyCompanyAttribute), False)
                ' If there aren't any Company attributes, return an empty string
                If attributes.Length = 0 Then
                    Return ""
```

```
                End If
                ' If there is a Company attribute, return its value
                Return (CType(attributes(0), AssemblyCompanyAttribute)).Company
            End Get
        End Property
        #End Region
End Class
```

Pantalla Aprender

Pantalla que me permite ingresar nuevo conocimiento en el sistema

```
Libreria para manipular la sintetizacion de voz
Imports SpeechLib
Libreria que me permite manipular la base de datos
Imports MySql.Data.MySqlClient

Public Class Aprender
   Atributo que me permite manipular la base de datos
   Private conexion As New MySqlConnection
'creamos una variable para las consultas
Private consultas As String
   Private comandos As New MySqlCommand
Propiedad que me permite manipular el reconocimiento de voz
   Private WithEvents engSR As New System.Speech.Recognition.SpeechRecognizer
Propiedad que me permite manipular la sintetizacion de voz
   Private WithEvents engSS As New System.Speech.Synthesis.SpeechSynthesizer

Procedimiento que libera libera los recursos utilizados en esta pantalla
   Private Sub Aprender_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
```

```
        Dispose()
        engSR.Dispose()
    End Sub
```

Procedimiento que me permite guardar toda la informacion del Nuevo conocimiento

```
Public Sub grabacion()
If (Me.t_tema.Text.Trim = "") Then
            habla.guardar("Ingrese el Tema Por Favor", "tempo1.wma")
            DsAudioPlayer1.FileName = "c:\SEUZ\tempo1.wma"
            DsAudioPlayer1.Start()
            Exit Sub
        End If
        If (Me.contenido.Text.Trim = "") Then
            habla.guardar("Ingrese el contenido por favor", "tempo3.wma")
            DsAudioPlayer1.FileName = "c:\SEUZ\tempo3.wma"
            DsAudioPlayer1.Start()
            Exit Sub
        End If
        If (Me.fuente.Text.Trim = "") Then
            habla.guardar("Ingrese la Fuente por favor", "tempo2.wma")
            DsAudioPlayer1.FileName = "c:\SEUZ\tempo2.wma"
            DsAudioPlayer1.Start()
            Exit Sub
        End If
        If (Me.observacion.Text.Trim = "") Then
            habla.guardar("Ingrese las observaciones por favor", "tempo4.wma")
            DsAudioPlayer1.FileName = "c:\SEUZ\tempo4.wma"
            DsAudioPlayer1.Start()
            Exit Sub
        End If
```

```
        infor.tema = Me.t_tema.Text.ToLower
        infor.fuente = Me.fuente.Text
        infor.contenido = Me.contenido.Text
        infor.observacion = Me.observacion.Text
        infor.guardar(usuario.id, Me.combo_categoria.SelectedValue)
        Me.fuente.Text = ""
        Me.observacion.Text = ""
        Me.t_tema.Text = ""
        Me.contenido.Text = ""
        Me.t_tema.Focus()    End Sub
```

Procedimiento que me permite cargar las categoria de los conocimientos y commandos en este sistema.

```
Private Sub Aprender_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        conexion.ConnectionString =
"server=localhost;userid=root;password=dar;database=seuz"
'Abrimos la conexión y comprobamos que no hay error
conexion.Open()
        consultas = "select * from seuz.categoria"
comandos.Connection = conexion
        comandos.CommandText = consultas
        'Como el comando no es almacenado ni vinculado a un tabla elegimos el tipo .Text
        comandos.CommandType = CommandType.Text
        'Creamos un lector de datos (IDataReader) y lo inicializamos
        'con el lector del objeto MySQLCommand
Dim dr As New MySqlDataAdapter
        Dim datos As New DataSet
dr.SelectCommand = comandos
        dr.Fill(datos)
```

```
        'combinamos las palabras con el conocimiento
        Me.combo_categoria.DataSource = datos.Tables(0)
        Me.combo_categoria.ValueMember = "id"
        Me.combo_categoria.DisplayMember = "descripcion"
        Me.t_usuario.Text = usuario.Nombre
        Me.ThemeName = confi.tema
        Me.t_tema.ThemeName = confi.tema
        Me.t_usuario.ThemeName = confi.tema
        Me.Cerrar.ThemeName = confi.tema
Me.combo_categoria.ThemeName = confi.tema
            Me.fuente.ThemeName = confi.tema
            Me.observacion.ThemeName = confi.tema
            Me.b_Aceptar.ThemeName = confi.tema
            Me.fecha.ThemeName = confi.tema
            ' creo los comandos con los que voy a trabajar
Dim itemA As New Speech.Recognition.SrgsGrammar.SrgsItem("tema")
            Dim itemB As New Speech.Recognition.SrgsGrammar.SrgsItem("categoria")
            Dim itemC As New Speech.Recognition.SrgsGrammar.SrgsItem("contenido")
            Dim itemD As New Speech.Recognition.SrgsGrammar.SrgsItem("fuente")
            Dim itemE As New Speech.Recognition.SrgsGrammar.SrgsItem("observacion")
        Dim itemF As New Speech.Recognition.SrgsGrammar.SrgsItem("grabar")
        Dim itemG As New Speech.Recognition.SrgsGrammar.SrgsItem("cancelar")
‘creo la regla con los comandos que tengo q incorporar
Dim rule As New Speech.Recognition.SrgsGrammar.SrgsRule("Comandos", _
              New Speech.Recognition.SrgsGrammar.SrgsOneOf(itemA, itemB, itemC, itemD,
itemE, itemF))
‘creo el documento de ejemplo
Dim doc As New Speech.Recognition.SrgsGrammar.SrgsDocument
‘agrego la regla y la pongo como regla por defecto
doc.Rules.Add(rule)
```

```
        doc.Root = rule
‘cargo la gramatica de la aplicacion
engSR.LoadGrammar(New Speech.Recognition.Grammar(doc))
   End Sub

Procedimiento que me permite manipular el reconocimiento de voz
   Private Sub engSR_SpeechRecognized(ByVal sender As Object, ByVal e As
System.Speech.Recognition.SpeechRecognizedEventArgs) Handles
engSR.SpeechRecognized
      Select Case e.Result.Text
         Case "grabar"
grabacion()
         Case ("tema")
            Me.t_tema.Focus()
         Case "contenido"
            Me.contenido.Focus()
         Case "categoria"
            Me.combo_categoria.Focus()
         Case "fuente"
            Me.fuente.Focus()
         Case "observacion"
            Me.observacion.Focus()
Case "cancelar"
            Me.Close()
      End Select
   End Sub

Procedimiento que me permite cerrar esta pantalla
   Private Sub Cerrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cerrar.Click
```

```
        Me.Close()
    End Sub

Procedimiento que me permite grabar la informacion ingresada en esta pantalla
Private Sub b_Aceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles b_Aceptar.Click
        grabacion()
    End Sub
End Class

Diseño
Este codigo se genera automaticamente cuando diseño de esta pantalla
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Aprender
    Inherits Telerik.WinControls.UI.RadForm
    'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
```

```
    'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.t_usuario = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel1 = New Telerik.WinControls.UI.RadLabel()
        Me.l_tema = New Telerik.WinControls.UI.RadLabel()
        Me.t_tema = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel2 = New Telerik.WinControls.UI.RadLabel()
        Me.combo_categoria = New Telerik.WinControls.UI.RadDropDownList()
        Me.fecha = New Telerik.WinControls.UI.RadDateTimePicker()
        Me.RadLabel3 = New Telerik.WinControls.UI.RadLabel()
        Me.RadLabel4 = New Telerik.WinControls.UI.RadLabel()
        Me.b_Aceptar = New Telerik.WinControls.UI.RadButton()
        Me.Cerrar = New Telerik.WinControls.UI.RadButton()
        Me.RadLabel5 = New Telerik.WinControls.UI.RadLabel()
        Me.fuente = New Telerik.WinControls.UI.RadTextBox()
        Me.RadLabel6 = New Telerik.WinControls.UI.RadLabel()
        Me.observacion = New Telerik.WinControls.UI.RadTextBox()
        Me.contenido = New System.Windows.Forms.RichTextBox()
        CType(Me.t_usuario, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.l_tema, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.t_tema, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.combo_categoria, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.fecha, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel3, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.b_Aceptar, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.Cerrar, System.ComponentModel.ISupportInitialize).BeginInit()
```

```
        CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.fuente, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.RadLabel6, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.observacion, System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        't_usuario
        '
        Me.t_usuario.Enabled = False
        Me.t_usuario.Location = New System.Drawing.Point(73, 12)
        Me.t_usuario.Name = "t_usuario"
        Me.t_usuario.Size = New System.Drawing.Size(200, 20)
        Me.t_usuario.TabIndex = 0
        Me.t_usuario.TabStop = False
        '
        'RadLabel1
        '
        Me.RadLabel1.Location = New System.Drawing.Point(12, 16)
        Me.RadLabel1.Name = "RadLabel1"
        Me.RadLabel1.Size = New System.Drawing.Size(45, 16)
        Me.RadLabel1.TabIndex = 1
        Me.RadLabel1.Text = "Usuario"
        Me.RadLabel1.ThemeName = "Desert"
        '
        'l_tema
        '
        Me.l_tema.Location = New System.Drawing.Point(12, 59)
    Me.l_tema.Name = "l_tema"
    Me.l_tema.Size = New System.Drawing.Size(35, 16)
```

```
Me.l_tema.TabIndex = 2
        Me.l_tema.Text = "Tema"
Me.l_tema.ThemeName = "Desert"
        '
        't_tema
        '
        Me.t_tema.Location = New System.Drawing.Point(73, 55)
        Me.t_tema.Name = "t_tema"
        Me.t_tema.Size = New System.Drawing.Size(288, 20)
        Me.t_tema.TabIndex = 3
        Me.t_tema.TabStop = False
        '
        'RadLabel2
        '
        Me.RadLabel2.Location = New System.Drawing.Point(12, 101)
        Me.RadLabel2.Name = "RadLabel2"
        Me.RadLabel2.Size = New System.Drawing.Size(55, 16)
        Me.RadLabel2.TabIndex = 4
        Me.RadLabel2.Text = "Categoria"
        Me.RadLabel2.ThemeName = "Desert"
        '
        'combo_categoria
        '
        Me.combo_categoria.FilterExpression = Nothing
        Me.combo_categoria.Location = New System.Drawing.Point(73, 97)
        Me.combo_categoria.Name = "combo_categoria"
        Me.combo_categoria.Size = New System.Drawing.Size(204, 20)
        Me.combo_categoria.TabIndex = 6
        Me.combo_categoria.Text = "RadDropDownList1"
'
```

```
        'fecha
        '
        Me.fecha.Culture = New System.Globalization.CultureInfo("es-ES")
Me.fecha.Enabled = False
        Me.fecha.Format = System.Windows.Forms.DateTimePickerFormat.[Long]
        Me.fecha.Location = New System.Drawing.Point(73, 143)
        Me.fecha.MaxDate = New Date(9998, 12, 31, 0, 0, 0, 0)
        Me.fecha.MinDate = New Date(1900, 1, 1, 0, 0, 0, 0)
        Me.fecha.Name = "fecha"
        Me.fecha.NullDate = New Date(1900, 1, 1, 0, 0, 0, 0)
        Me.fecha.Size = New System.Drawing.Size(195, 20)
        Me.fecha.TabIndex = 7
        Me.fecha.TabStop = False
        Me.fecha.Text = "RadDateTimePicker1"
        Me.fecha.Value = New Date(2010, 11, 30, 0, 0, 0, 0)
        '
        'RadLabel3
        '
        Me.RadLabel3.Location = New System.Drawing.Point(12, 143)
        Me.RadLabel3.Name = "RadLabel3"
        Me.RadLabel3.Size = New System.Drawing.Size(38, 16)
        Me.RadLabel3.TabIndex = 8
        Me.RadLabel3.Text = "Fecha"
        Me.RadLabel3.ThemeName = "Desert"
        '
        'RadLabel4
        '
        Me.RadLabel4.Location = New System.Drawing.Point(14, 191)
        Me.RadLabel4.Name = "RadLabel4"
        Me.RadLabel4.Size = New System.Drawing.Size(58, 16)
```

```
        Me.RadLabel4.TabIndex = 9
        Me.RadLabel4.Text = "Contenido"
        Me.RadLabel4.ThemeName = "Desert"
        '
        'b_Aceptar
        '
        Me.b_Aceptar.Location = New System.Drawing.Point(192, 533)
Me.b_Aceptar.Name = "b_Aceptar"
Me.b_Aceptar.Size = New System.Drawing.Size(99, 30)
Me.b_Aceptar.TabIndex = 11
        Me.b_Aceptar.Text = "Aceptar"
'
        'Cerrar
        '
        Me.Cerrar.Location = New System.Drawing.Point(408, 533)
        Me.Cerrar.Name = "Cerrar"
        Me.Cerrar.Size = New System.Drawing.Size(99, 30)
Me.Cerrar.TabIndex = 12
        Me.Cerrar.Text = "Cerrar"
'
        'RadLabel5
        '
        Me.RadLabel5.Location = New System.Drawing.Point(14, 385)
        Me.RadLabel5.Name = "RadLabel5"
        Me.RadLabel5.Size = New System.Drawing.Size(40, 18)
Me.RadLabel5.TabIndex = 13
        Me.RadLabel5.Text = "Fuente"
        '
        'fuente
'
```

```
        Me.fuente.Location = New System.Drawing.Point(61, 385)
        Me.fuente.Multiline = True
        Me.fuente.Name = "fuente"
        '
        '
        '
        Me.fuente.RootElement.StretchVertically = True
        Me.fuente.Size = New System.Drawing.Size(446, 33)
        Me.fuente.TabIndex = 14
        Me.fuente.TabStop = False
        '
        'RadLabel6
        '
        Me.RadLabel6.Location = New System.Drawing.Point(14, 424)
        Me.RadLabel6.Name = "RadLabel6"
        Me.RadLabel6.Size = New System.Drawing.Size(82, 16)
Me.RadLabel6.TabIndex = 15
        Me.RadLabel6.Text = "Observaciones"
Me.RadLabel6.ThemeName = "Desert"
        '
        'observacion
        '
        Me.observacion.Location = New System.Drawing.Point(102, 424)
Me.observacion.Multiline = True
        Me.observacion.Name = "observacion"
'
        '
        '
        Me.observacion.RootElement.StretchVertically = True
        Me.observacion.Size = New System.Drawing.Size(422, 103)
```

```
Me.observacion.TabIndex = 16
        Me.observacion.TabStop = False
'
        'contenido
        '
        Me.contenido.Location = New System.Drawing.Point(78, 179)
        Me.contenido.Name = "contenido"
        Me.contenido.Size = New System.Drawing.Size(429, 200)
Me.contenido.TabIndex = 17
        Me.contenido.Text = ""
'
        'Aprender
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(536, 575)
Me.Controls.Add(Me.contenido)
        Me.Controls.Add(Me.observacion)
Me.Controls.Add(Me.RadLabel6)
Me.Controls.Add(Me.fuente)
Me.Controls.Add(Me.RadLabel5)
Me.Controls.Add(Me.Cerrar)
        Me.Controls.Add(Me.b_Aceptar)
Me.Controls.Add(Me.RadLabel4)
        Me.Controls.Add(Me.RadLabel3)
Me.Controls.Add(Me.fecha)
        Me.Controls.Add(Me.combo_categoria)
        Me.Controls.Add(Me.RadLabel2)
        Me.Controls.Add(Me.t_tema)
        Me.Controls.Add(Me.l_tema)
```

```
Me.Controls.Add(Me.RadLabel1)
        Me.Controls.Add(Me.t_usuario)
        Me.MaximizeBox = False
Me.Name = "Aprender"
        '
        '
        '
Me.RootElement.ApplyShapeToControl = True
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "-"
        Me.TopMost = True
        CType(Me.t_usuario, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.l_tema, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.t_tema, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.combo_categoria, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.fecha, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel3, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel4, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.b_Aceptar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.Cerrar, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel5, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.fuente, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.RadLabel6, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.observacion, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()
    End Sub
```

```
    Friend WithEvents t_usuario As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel1 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents l_tema As Telerik.WinControls.UI.RadLabel
    Friend WithEvents t_tema As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel2 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents combo_categoria As Telerik.WinControls.UI.RadDropDownList
    Friend WithEvents fecha As Telerik.WinControls.UI.RadDateTimePicker
    Friend WithEvents RadLabel3 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents RadLabel4 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents b_Aceptar As Telerik.WinControls.UI.RadButton
    Friend WithEvents Cerrar As Telerik.WinControls.UI.RadButton
    Friend WithEvents RadLabel5 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents fuente As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents observacion As Telerik.WinControls.UI.RadTextBox
    Friend WithEvents RadLabel6 As Telerik.WinControls.UI.RadLabel
    Friend WithEvents contenido As System.Windows.Forms.RichTextBox
End Class
```

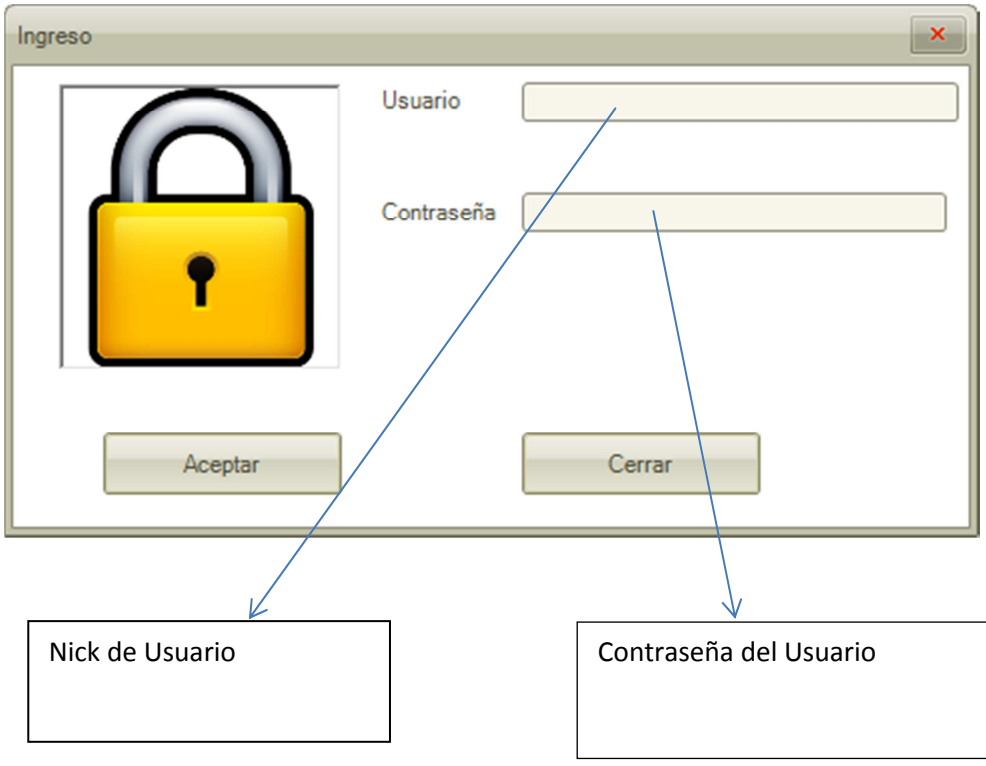
MANUAL DE USUARIO

Pantalla de Bienvenida

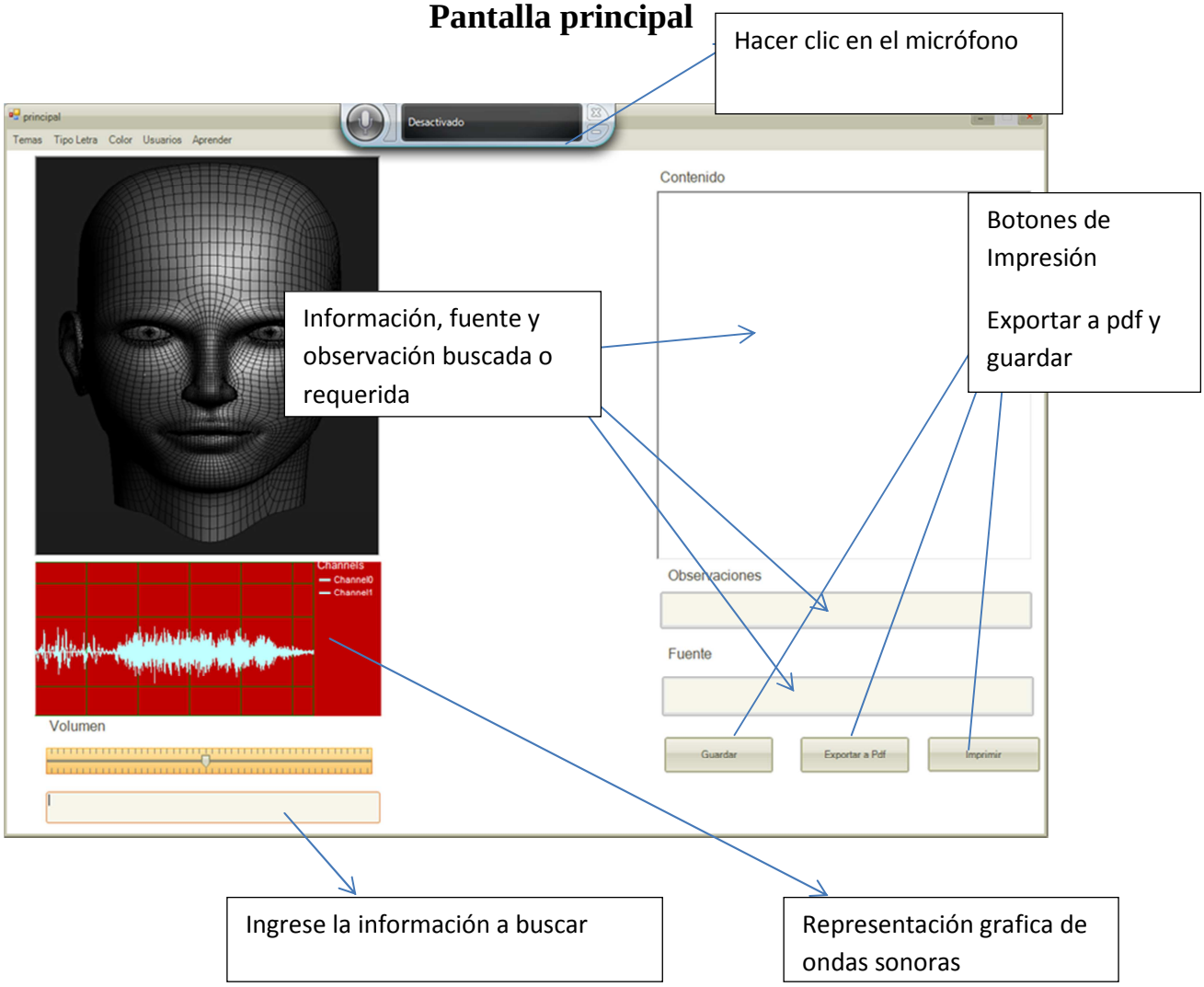
Esperar que llegue la barra de progreso hasta el 100 % y da hay sale la pantalla de login.

Pantalla de Login

Es esta pantalla se ingresa el Nick y la clave del usuario que va a interactuar con el Usuario si el usuario existe en el sistema pasa a la pantalla principal. El usuario solo tiene tres intentos para ingresar al sistema después de eso se cerrar la pantalla.



Pantalla principal



Este sistema tiene dos maneras de interactuar con el usuario descriptas como modo.

Modo1

1- Primero hacer clic en el micrófono o si esta configurado correctamente el micrófono lo puede activar por comando de voz diciendo “Activar microfono”

2- En esta parte puede ver la información que se desea buscar luego si el sistema reconoce el comando te dirá buscando información y después de encontrar la información se llenará los cuadro de texto contenido, observación fuente y se procederá a reproducir la información a manera de archivo de sonido y representado gráficamente las ondas sonoras.

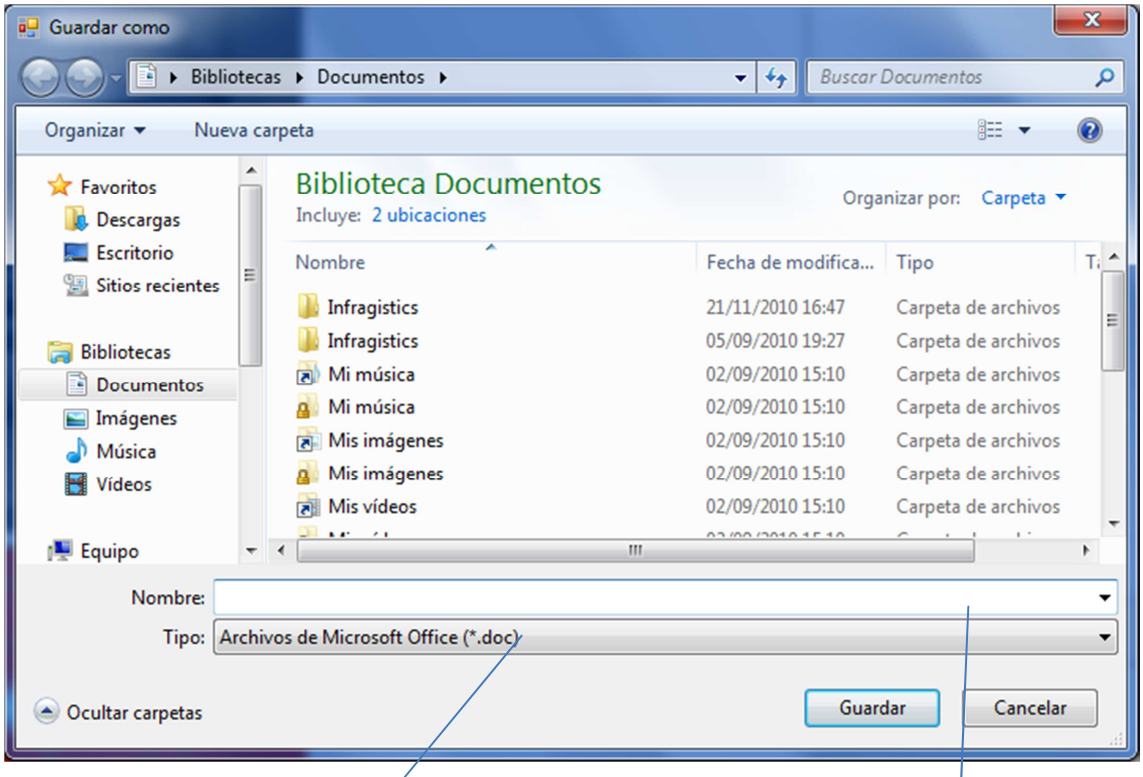
3- Si ya no desea escuchar la información puede decir el comando parar.
4- Si se desea imprimir puede dictar el comando imprimir y se abrirá automáticamente la pantalla de impresión.
5- Si desea guardar la información en formato pdf puede decir el comando exportar a pdf
6- Si desea guardar la información en formato Word 2003,2007 y archivo texto decir el comando exportar a seguido del tipo de archivo que desea guardar.

Modo2

1. Puede ingresar manualmente la información a buscar luego esperar hasta que se llene el cuadro contenido, fuente y observaciones
2. Si desea imprimir la información haga clic en el botón imprimir
3. Si desea guardar la información en archivo pdf haga clic en el botón exportar a pdf
4. Si desea guardar la información haga clic en el botón guardar y te saldrá la pantalla guardar como

Pantalla de Guardar Como

En esta pantalla podrá ingresar el nombre del archivo y escoger el tipo de archivo que desea guardar, luego de haber ingresado el nombre haga clic en guardar.

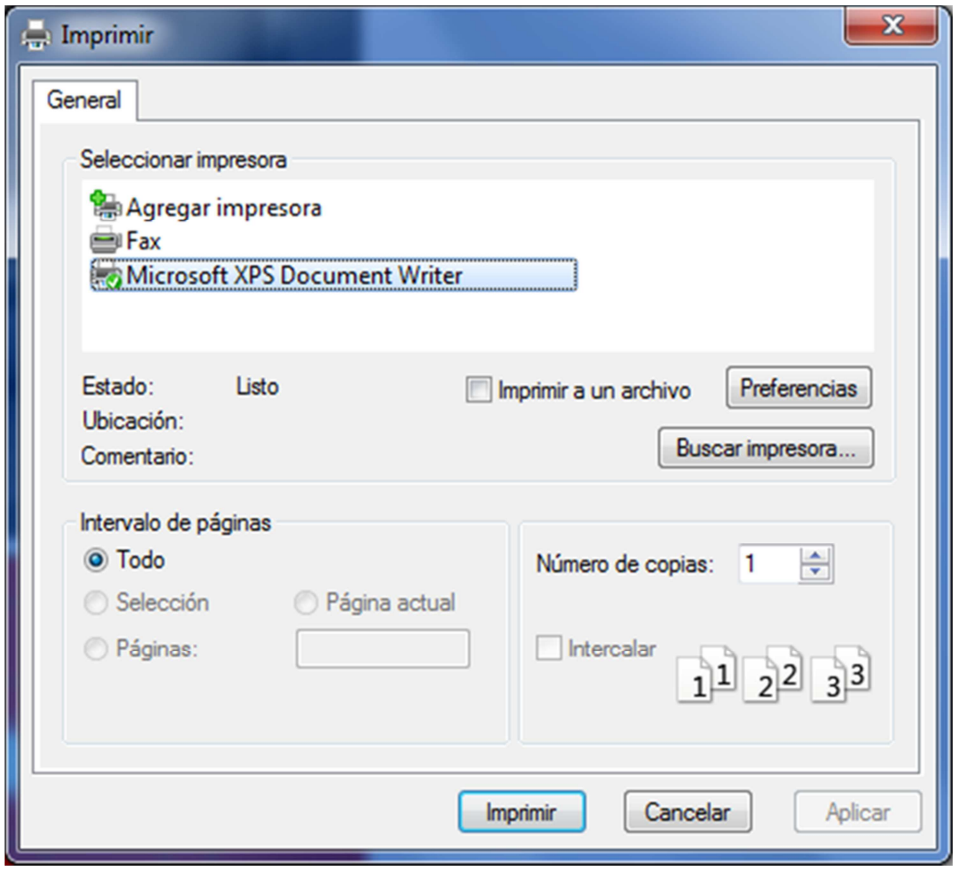


Tipo de archivo

Nombre del archivo

Pantalla de Impresión

En esta pantalla simplemente podrá configurar la impresora a su elección.



Pantalla de Tipo de letra

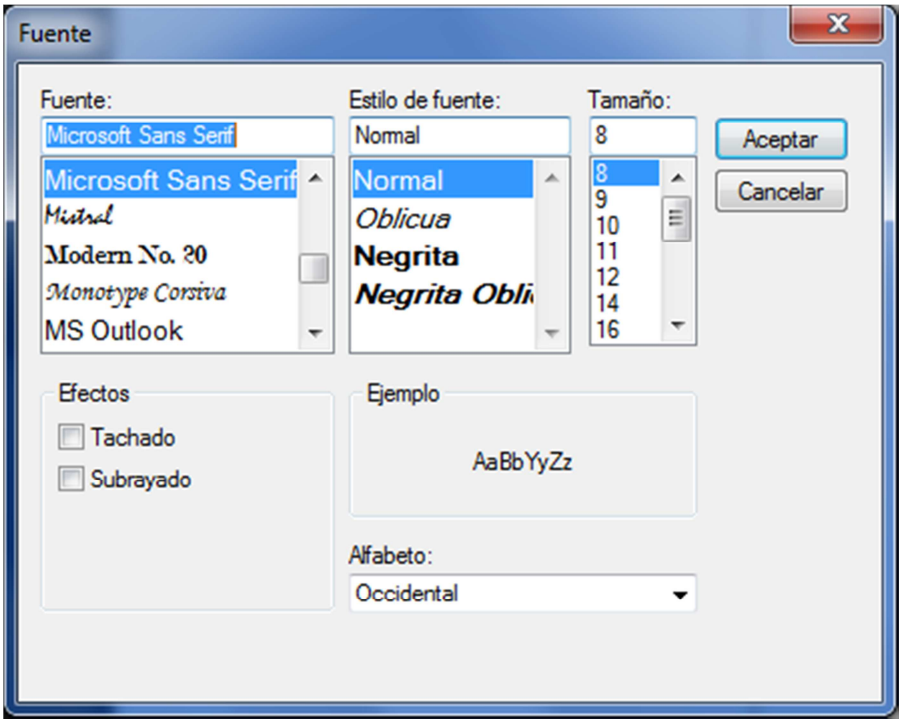
Modo 1

Puede ingresar a esta venta dictando el comando fuente

Modo 2

Puede ingresar a esta ventana haciendo clic en el menú principal y luego ir a fuente

En esta pantalla podrá escoger el tipo de letra con el que se va a imprimir la información.



Pantalla de Color de Texto

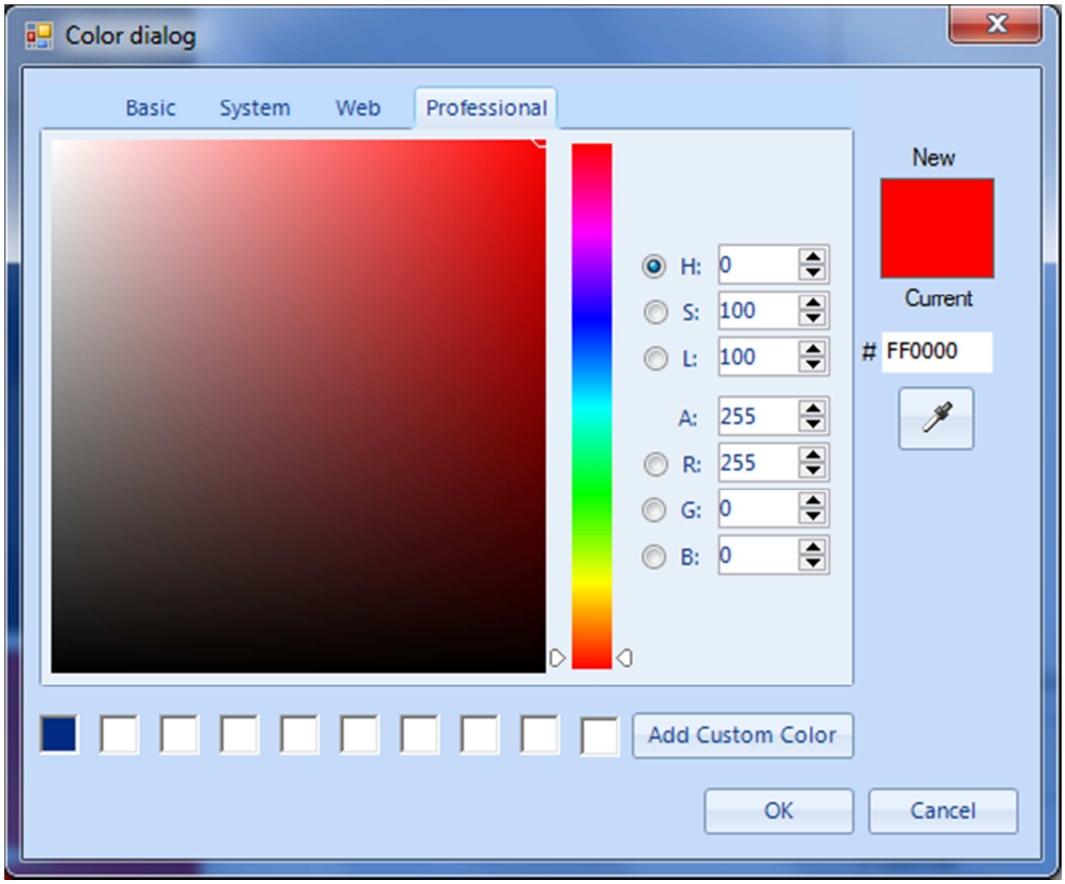
Modo 1

Puede ingresar a esta ventana diciendo el comando color.

Modo 2

Puede ingresar a esta ventana haciendo clic en el menú principal y luego ir a color

En esta pantalla podrá escoger el color con la que va a ser impresa la información.



Pantalla de Temas

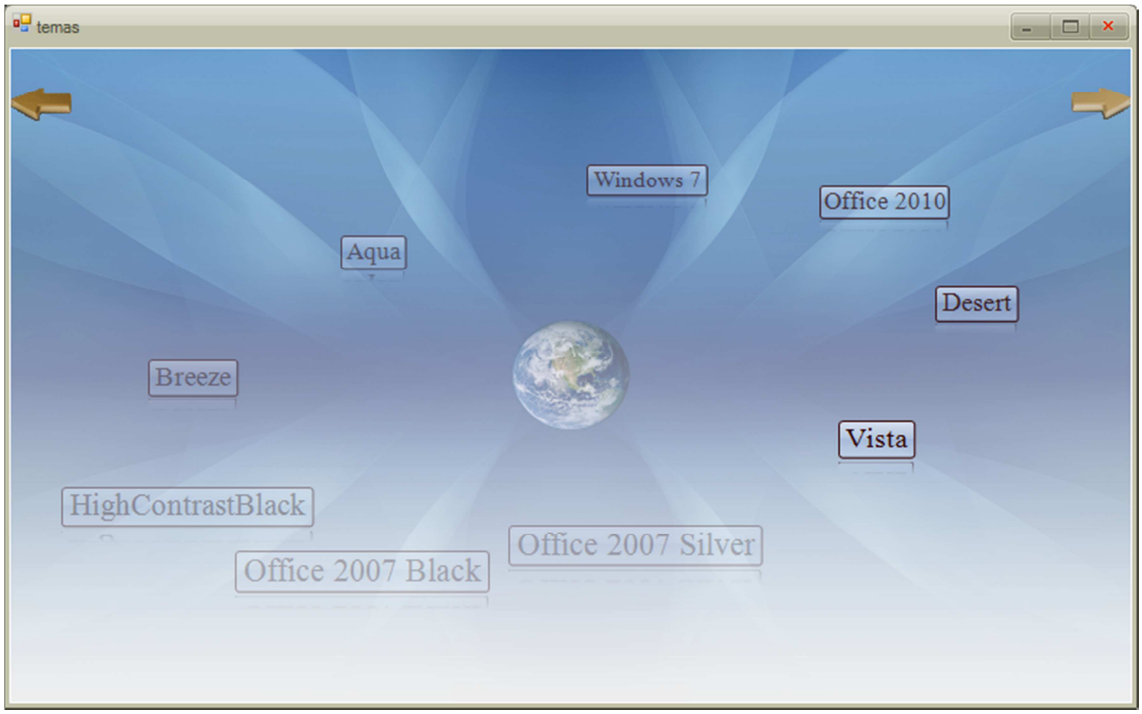
Modo 1

Puede ingresar a esta ventana dictando el comando temas

Modo 2

Puede ingresar a esta ventana haciendo clic en el menú principal y luego ir a temas

En esta pantalla podrá escoger los 9 diferentes temas que tienes el sistema ya sea haciendo clic en el tema específico o diciendo el nombre del tema ejemplo aqua.



Pantalla de Ingreso de Nuevo Usuario

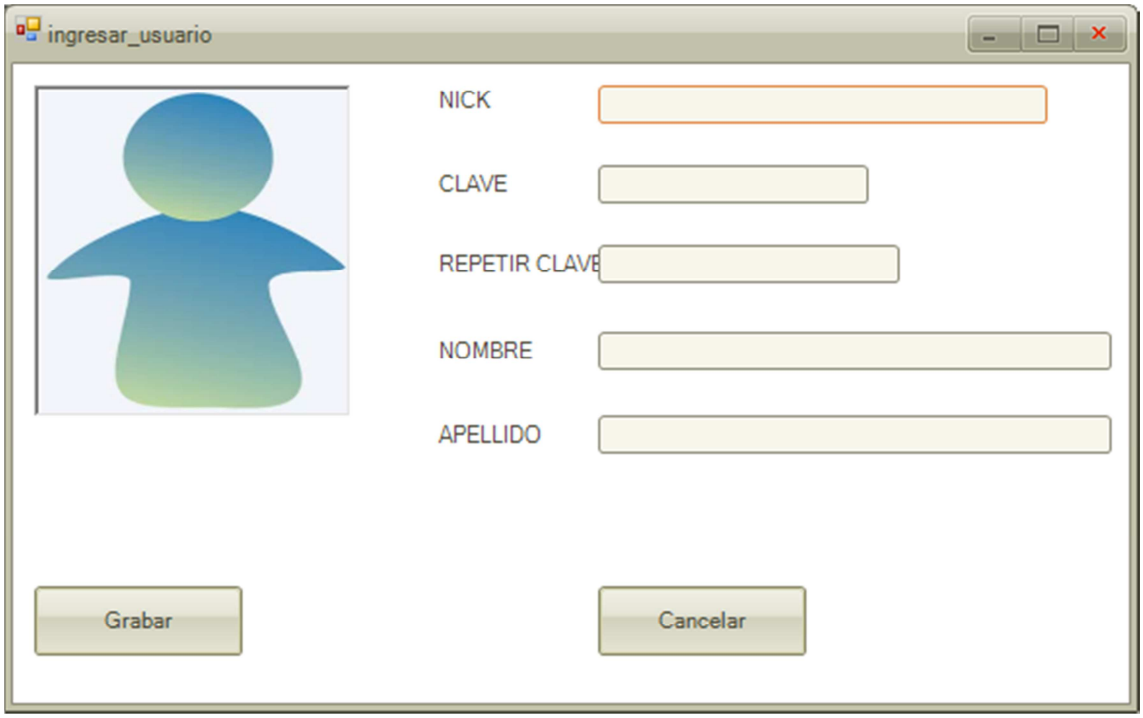
Modo 1

Puede ingresar a esta pantalla diciendo al comando nuevo usuario

Modo 2

Puede ingresar a esta ventana haciendo clic en el menú principal y luego ir a usuario y nuevo usuario.

En esta pantalla podrá ingresar los datos del nuevo usuario y luego de ingresar todo la información requerida haga clic en grabar o decir el comando grabar.



Pantalla de Ingreso de Conocimiento

Modo 1

Para ingresar a esta pantalla puede decir el comando Aprender y para interactuar con esta pantalla por ejemplo si desea ingresar el contenido diga el comando contenido y así sucesivamente y luego para guardar puede decir el comando grabar.

Modo 2

Puede ingresar a esta ventana haciendo clic en el menú principal y luego ir a Aprender. Luego de haber ingresado toda la información requerida puede hacer clic en aceptar.